NPS67-79-007

NAVAL POSTGRADUATE SCHOOL

Monterey, California

A General Procedure for Obtaining Velocity Vector from a System of High Response Impact Pressure Probes

D. Adler and R. P. Shreeve

July 1979

Prepared for:
Naval Air Systems Command
[illegible]nington, DC
Office of Naval Research
Arlington, VA
[illegible]and

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral T. F. Dedman
Superintendent

Jack R. Borsting
Provost

The work reported herein was supported by the Naval Air Systems Command, Washington, DC. and Office of Naval Research, Arlington, VA.

This report was prepared by:

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
NPS67-79-007		

4. TITLE (and Subtitle)
A General Procedure for Obtaining Velocity Vector from a System of High Response Impact Pressure Probes

5. TYPE OF REPORT & PERIOD COVERED
Technical Report
July 1978 - July 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
D. Adler and R. P. Shreeve

8. CONTRACT OR GRANT NUMBER(s)
61153N;
N00019-79-WR-91115

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, CA 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Air Systems Command
Washington, DC 20361

12. REPORT DATE
July 1979

13. NUMBER OF PAGES
~~78~~ 84

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for Public Release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Flow Measurements

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A technique to measure a high frequency, repetitively pulsating flow filed is presented. Two impact tube probes and a Kiel probe, all with pressure transducers mounted in their tips are used. Five readings are required to identify a velocity vector at a point.
In this report the technique, the numerical procedure and the computer program used are described.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A General Procedure for Obtaining Velocity Vector From a System of High Response Impact Pressure Probes

by

D. Adler and R. P. Shreeve

Contents

Notation

C_p - Pressure coefficient defined in eq. (3)

α - Yaw angle relative to laboratory space

ϕ - Pitch angle relative to laboratory space

P_t - Total pressure

P_s - Static Pressure

α_{rp} - Yaw angle relative to probe axis

ϕ_{rp} - Pitch angle relative to probe axis

P_p - Pressure indicated by the probe

α_p - Yaw setting of the probe

ϕ_p - Pitch setting of the probe

Subscripts

I, II, III, IV } relating to probe positions shown in figure 3

A relating to A type probe

B relating to B type probe

Lo lowest value in an array

rp relative to the probe axis

Up highest value in an array

Notation of PENPTS

DD - denominator in the subsequent line of the program

HT - penetration height (Cp of a calibration surface)

IR - a flag with a value of either 1 or 2 to identify first or second penetration point. This flag also controls return of values to calling program or subroutine

IB - a flag indicating the location of the last rectangle in the band checked for a penetration point. It prevents repetition of the scanning in the 103 Do loop

NX - number of X values in the calibration surface grid (≥ values of the calibration surface)

NY - number of Y values in the calibration surface grid (values of the calibration surface)

X(I) - X values in the grid

XM - value of X at point B (fig. 5)

XR(I) - two values of X returned by the subroutine as results

XS - currently calculated value of X which is the present result and is later stored in XR(I)

Y(J) - Y values in the calibration grid

YC - intermediate value used in the subsequent line of the program

YG - the Y position of the penetrating line*

Z(I.J)- Values of the Z = Cp calibration surface above grid points

ZX - value of Z either at point A or point C (see figs. 5, 6 and 7)

ZM - value of Z at point B

---

* It should be pointed out here that the penetrating line is parallel to the X axis at a height HT = Cp above the X,Y plane and at a distance YG from the X axis on the X,Y plane

Notation of INTSCS

AA	-	constant defined in eq. 8
AB	-	constant defined in eq. 9
ARIN	-	ordinate at which scanning starts
ARLO	-	lowest ordinate of calibration surface (as a condition it must be identical for all calibration surfaces used)
ARM1	-	ordinate measured relative to probe 1 (the probe belonging to the first curve Cp = const.)
ARM2	-	ordinate measured relative to probe 2 (the probe belonging to the second curve Cp = const)
ARP1	-	setting ordinate of probe 1
ARP2	-	setting ordinate of probe 2
ARRES (1)	-	abscissa value returned to INTSCS from PENPTS
ARUP	-	highest ordinate of calibration surface (as a condition it must be identical for all calibration surfaces used)
AR2R	-	ordinate relative to laboratory space
AR2RJ	-	ordinate relative to laboratory space of previous scan
BA	-	constant defined in eq. 8
BB	-	constant defined in eq. 9
DAR	-	ordinate step for scanning
HT1	-	penetration height of calibration surface of probe 1 (Cp of probe 1)
HT2	-	penetration height of calibration surface of probe 2 (Cp of probe 2)
IEPS	-	a flag which is equal to 2 when the calculation is carried out as shown in fig. 11, otherwise it is equal to 1. (This information is required in the main program and is returned to it)

ISL - a flag which is equal to 10 if one of the left hand side results returned from PENPTS is 1000.0 (i.e., is not a penetration point). Otherwise it is equal to 1

ISR - a flag which is equal to 10 if one of the right hand side results returned from PENPTS is 1000.0 (i.e., is not a penetration point). Otherwise it is equal to 1

K - a flag indicating whether the first or the second intersection is evaluated

N1 - number of abscissas in calibration surface grid (fig. 4)

N2 - number of Ordinates in calibration surface grid (fig. 4)

RES1(K) - evaluated and returned abscissa of intersection

RES2(K) - evaluated and returned ordinate of intersection

R1 - radius in fig. 11 for probe 1

R2 - radius in fig. 11 for probe 2

RAIL - abscissa of point AIL in fig. 9 or 10

RAIR - abscissa of point AIR in fig. 9 or 10

RAJL - abscissa of point AJL in fig. 9 or 10

RAJR - abscissa of point AJR in fig. 9 or 10

RBIL - abscissa of point BIL in fig. 9 or 10

RBIR - abscissa of point BIR in fig. 9 or 10

RBJL - abscissa of point BJL in fig. 9 or 10

RBJR - abscissa of point BJR in fig. 9 or 10

X1(I) - abscissas of calibration surface of probe 1 or its transformation as determined by calling statement in main program

X2(I) - abscissas of calibration surface of probe 2 or its transformation as determined by calling statement in main program

X3(I) - abscissas of calibration surface of probe 1 or its transformation as determined by calling statement in main program

X4(I) - abscissas of calibration surface of probe 2 or its transformation as determined by calling statement in main program

Y1(I) - ordinates matching X1(I)

Y2(I) - ordinates matching X2(I)

Y3(I) - ordinates matching X3(I)

Y4(I) - ordinates matching X4(I)

Z1(I,J) - values of Cp_{rp} for X1(I) and Y1(I)

Z2(I,J) - values of Cp_{rp} for X2(I) and Y2(I)

Z3(I,J) - values of Cp_{rp} for X3(I) and Y3(I)

Z4(I,J) - values of Cp_{rp} for X4(I) and Y4(I)

Notation of Main Program VDR

A - variables in library routine IXCLOK used to evaluate computation times

ALF - values of α returned to VDR from INTSCS

ALFA - α input when experiment simulation is carried out

ALFC - α_{rpIII}

ALFCN - new value of α_{rpIII} for next iteration

ALFD - α_{rpIV}

ALFDN - new value of α_{rpIV} for next iteration

ALF1 - α_{I}

ALF2 - α_{II}

ALF3 - α_{I}

ALF4 - α_{III}

AM - mach number

CPA - Cp_{rpI}

CPB - Cp_{rpII}

CPC - Cp_{rpIII}

CPD - Cp_{rpIV}

EPSPS - relative difference between present static pressure and its value in the previous iteration

EPSPSG - convergence criterion (on the static pressure)

FALF - final α value (at convergence)

FPHI - final ϕ value (at convergence)

IA50 - a flag in IXCLOK

IA60 - a flag in IXCLOK

ICOPS - number of PS corrections carried out to ensure the Cp's are in calibration range

IEPS - a flag governing the convergence criterion returned from INTSCS

IEPS1 - a flag governing the convergence criterion for probes I and II

IEPS2 - a flag governing the convergence criterion for probes I and III

IIT - number of iterations on p_s

ISCAN - number of static pressure scans from initial static pressure guess upwards

IXCLOK - system subroutine for computation time evaluation

NOCOPS - a flag which when equal to 1 causes static pressure corrections to be carried out such that Cp's are always inside calibration range, and when equal to 2 cause these corrections to be skipped

NOSIM - a flag which when equal to 1 causes the program to simulate velocity measurement experiments and when equal to 2 causes the program to reduce measured data

NX - number of calibration surface matrix abscissa's, α values (must be identical for all calibration surface matrices used)

NY - number of calibration surface matrix ordinates, ϕ values (must be identical for all calibration surface matrices used)

PDYN - dynamic pressure

PHI - value of ϕ returned to VDR from INTSCS

PHIB - ϕ_{rpII}

PHII - ϕ input when experiment simulation is carried out

PHI1 - ϕ_{rpI}

PHI2 - ϕ_{rpII}

PHI3 - ϕ_{rpI}

PHI4 - ϕ_{rpIII}

PPA - signal of probe in position I

PPB - signal of probe in position II

PPC - signal of probe in position III

PPC - signal of probe in position IV

PS - static pressure

PSIN - initial guess of static pressure

PSN - new value of static pressure for next iteration

PSC - corrected static pressure

PST - memorized corrected static pressure

PT - total pressure measured

RELXPS - relaxation factor for convergence on the static pressure

VIRTIM - virtual computation time

XA - abscissas of calibration surface matrix of probe A

XAX - transformed abscissas of calibration surface matrix of probe A

XB - abscissas of calibration surface matrix of probe B

XBX - transformed abscissas of calibration surface matrix of probe B

XPC - yaw setting of position III

XPD - yaw setting of position IV

XRIN - α value at which scanning starts

XRLU - lowest α value of calibration surface (as a condition it must be identical for probes A and B)

XRUP - highest α value of calibration surface (as a condition it must be identical for probes A and B)

YA - ordinates of calibration surface matrix of probe A

YAX - transformed ordinates of calibration surface matrix of probe A

YB - ordinates of calibration surface matrix of probe B

YBX - transformed ordinates of calibration surface matrix of probe B

YPA - pitch setting of probe A

YPB - pitch setting of probe B

YRIN - ϕ value at which scanning starts

YRLO - lowest ϕ value of calibration surface (as a condition it must be identical for probes A and B)

YRUP - highest ϕ value of calibration surface (as a condition it must be identical for probes A and B)

ZA - Cp_{rpA} values of probe A

ZAX - transformed Cp_{rpA} values of probe A

ZAMAX - maximum of ZA array

ZAMIN - minimum of ZA array

ZB - Cp_{rpB} values of probe B

ZBX - transformed Cp_{rpB} values of probe B

ZBMAX - maximum of ZB array

ZBMIN - minimum of ZB array

Introduction

Experimental knowledge of the flow field generated by rotating turbo impellers is of prime importance in the research and development of turbomachinery. It is essential for the refinement of design methods, for the development of new flow models which include secondary flow and tip clearance effects, and particularly for the verification of new computer codes developed to calculate the flow through rotating blade rows.

In recent years laser velocimeter techniques have been applied successfully to measure the flow both inside and downstream of rotors. (Ref. 1 for example). It has become clear however, that the laser techniques are only reliable in the hands of experienced investigators. A window which remains clean is essential, and seeding is usually required. Laser techniques do not measure the pressure field and usually can only measure two components of the velocity unless the axis of the laser is tilted. Difficulty is also encountered when measuring close to walls. Hence there are reasons to consider alternative techniques, particularly if they are simpler to apply routinely in stationary turbomachinery passages. Furthermore, the achievement of redundancy in measuring the flow field behind the impeller is itself a worthwhile goal. The present

work deals with the application of a particular system of small high response pressures probes at the exit of an impeller.

Measurements behind an impeller, in the stationary bladeless gap, are simpler to make than measurements within the rotating passages. Transducer probes can be installed through the stationary machine casing and the data transmitted without resort to slip rings or rotary transformers. The sensor is not subjected to the centrifugal field or to the vibration of the rotor. However, the flow to be measured is then fluctuating at blade-passing frequency and any system of sensors must be calibrated for a wide range of possible Mach number, pitch angle, yaw angle and pressure variation - and yet must be capable of the necessary frequency response.

In Ref. 2, a method was described for using two semiconductor pressure probes together with the technique of synchronized sampling, to obtain the distribution of the velocity vector downstream of a rotor. The geometries of the two probes, designated Type A and Type B, and their installation in the compressor annulus are shown in Fig 1. It was argued that, in principle, by rotating the probes in yaw about their tips and controlling the sampling of the data from each probe to be at the same position in the rotor frame, the system of two separate probes could be used to acquire data at a point in the periodic flow

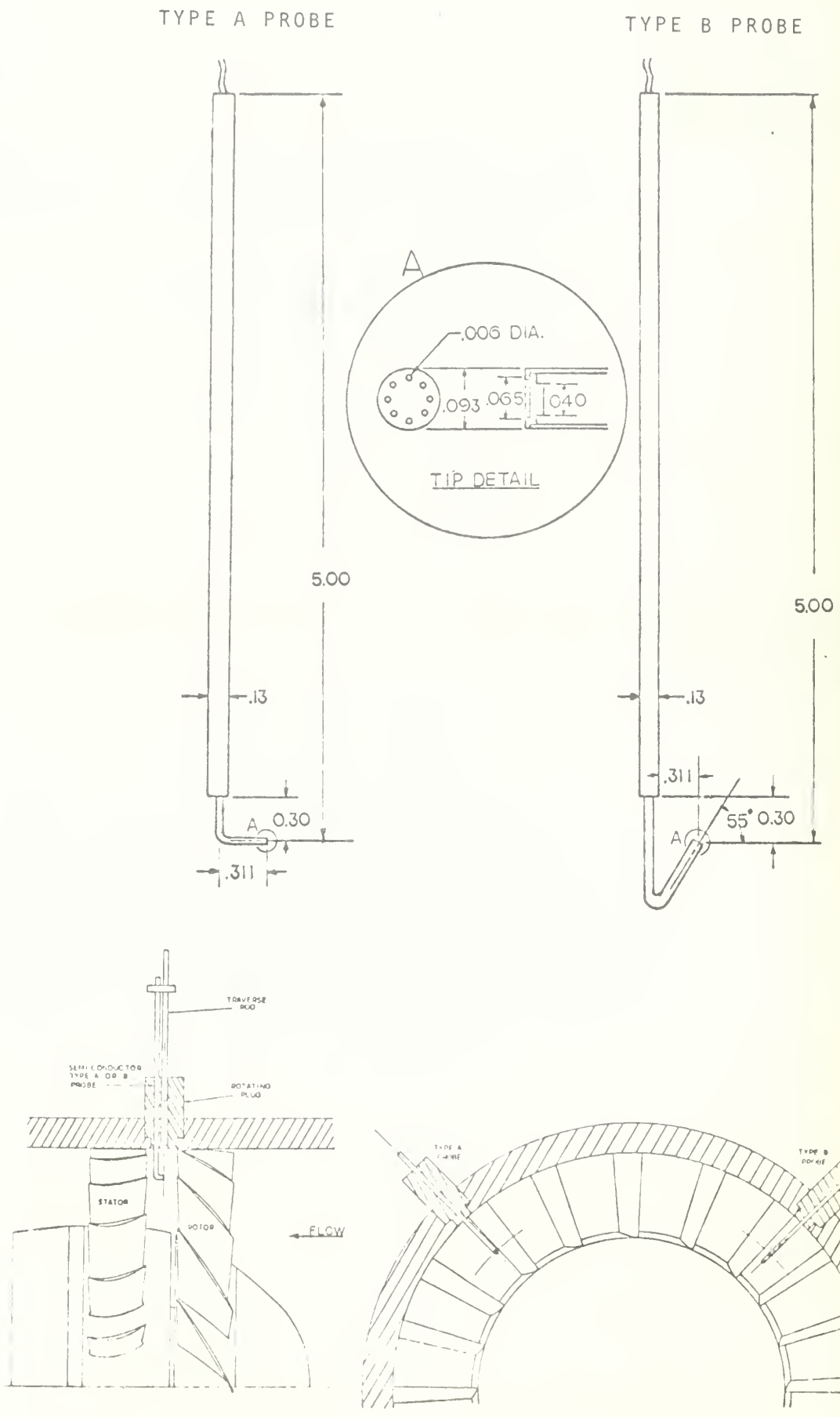


FIGURE 1. TYPE A AND TYPE B PROBES AND COMPRESSOR INSTALLATION.

from the rotor, corresponding to data normally obtained from the multiple sensors of four- or five hole pneumatic probes when measuring velocity in steady flows. The technique promised the use of probes having the simplest geometry and thus avoided the large size, expense and unreliability of multiple sensor probes which incorporate multiple semi-conductor transducers (Ref. 3). Because of the simple sensor tip geometry (that of a cylinder at incidence to the flow) the unsteady response was likely to be as good as could be expected of any single physical sensor.

The two-probe technique is strictly applicable only to periodic flows. However, data obtained on successive rotations of the rotor can be averaged to eliminate fluctuations which are not periodic. This was shown to be effective in tests reported in Ref. 2 in which a single Type A probe was used to establish the peripheral blade-to-blade distribution of flow yaw angle.

In order to obtain velocity from the pressure measurements which can be obtained from the two probes, the steady response characteristics must first be established in calibration tests carried out in a known, controlled, uniform flow. Second, a method of applying the calibration to measurements made in an unknown flow must be devised. In the present method two different approaches have been followed. In Ref. 4, a technique is given for representing

and applying the probe calibration analytically. When first applied the method gave surprisingly good accuracy (1 - 2%) since the method required that the probes had characteristics which could be well represented analytically. This in turn required that the probe tips be geometrically precise, a feature which was not present in the first generation of probes.

A second, more general approach is reported here, wherein the calibration of each probe is represented by a two-dimensional array of pressure coefficients. The application of the calibration given in this form, in an unknown flow required the development of special numerical procedures. The purpose of the present report is to document the analysis and the Fortran program developed to apply the method.

In its present form, the method does not require that the calibration "Surfaces" be symmetrical about any axis or be expressed in analytical form, but does require that the pressure coefficient be independent of Mach number. The latter restriction could undoubtedly be removed by introducing additional iterative steps. Further, in the present method only five measurements have to be taken to determine uniquely a velocity vector at a point. Throughout the report, the Fortran program notation has been used to describe the physics and equations involved in the solution.

Mathematical model

Assume the A and B type probes of ref. [1] (see also fig. 1) to be immersed in a three-dimensional steady flow field.* The pressure response of each of these probes in given gas is functionally described by four variables as:

$$P_p = P_p (\alpha, \phi, P_T, P_S) \tag{1}$$

If a pressure coefficient is defined as

$$Cp_{rp} = \frac{P_p - P_s}{P_T - P_s} \tag{2}$$

The calibration surface of each probe is given in the general case in form of a matrix of values of Cp, where

$$Cp_{rp} = Cp_{rp}(\alpha_{rp}, \phi_{rp}) \tag{3}$$

The pressure coefficient defined in this way has only a second order dependence on Mach and Reynolds numbers in the range of

---

* For our purpose, using the synchronized sampling, the flow field behind the impeller is steady, although the probes require a high speed response because of fluctuations.

$0 < Ma < 0.7$ in turbulent flows (Ref. 2), so that, to first order their influence is neglected in writing eq. (3).

If the type A probe is rotated about its axis into three different positions (i = I, III, IV) readings are taken, and the type B probe is fixed in position II and a single reading is taken, the following four equations can be written.

$$Cp_{Ai} = \frac{P_{pAi} - P_S}{P_T - P_S} = Cp_{Ai}(\alpha,\phi)$$
$$Cp_{BII} = \frac{P_{pBII} - P_S}{P_T - P_S} = Cp_{BII}(\alpha,\phi) \qquad (4)$$

it should be pointed out here, to avoid misunderstanding that α and ϕ are defined in a coordinate system relative to the machine axis and not relative to the probe axis. In the set of four equations (4) there are four unknown quantities, namely: α, ϕ, P_T and P_S. These are the quantities to be evaluated using the measured data. Together with the stagnation temperature they define the flow field uniquely. The four equations, resulting from the four measurements, should be sufficient to determine the four unknown quantities.

But the problem is complicated by three facts:

1) The calibration surfaces are not generally known in analytical form.
2) The calibration surfaces are double valued in α and ϕ i.e., for a given Cp and α there exist two ϕ values, or for a given Cp and ϕ there are two α values which satisfy eq. 3.

3) As a result of the measurement the value of P_p rather than of C_p is determined.

Since the calibration surfaces are not given in a simple analytical form the solution has to be numerical. An iterative procedure is required because P_p and not C_p values are measured. First P_s and P_T have to be guessed to yield C_p values, knowing the four measured values of P_p, and using eq. 4. The guess is then iteratively corrected to converge on the solution. However, convergence of the iterative procedure is complicated because of the double valued nature of the calibration surfaces.

This method of iteration shown in Fig. 2 was attempted initially for the evaluation of a measured point. In practice the initially suggested procedure converged in some cases and diverged in others, depending on the values, and signs of α and ϕ. This was not surprising as convergence on two variables is not likely to be a simple matter. However, in order that the measurement technique be useful, convergence on the correct solution for a general set of measurements, is absolutely necessary. In practice, this can certainly be achieved if one of the two iteration variables is obtained by measurement. Since the static pressure is a difficult quantity to measure even in a steady flow field, only the stagnation pressure measurement need be considered. It is possible that the time-varying stagnation pressure could be measured with a suitably designed Kiel probe. Data would then be taken by synchronized sampling from the fixed Kiel probe, from the type A probe rotated into two positions, and from the

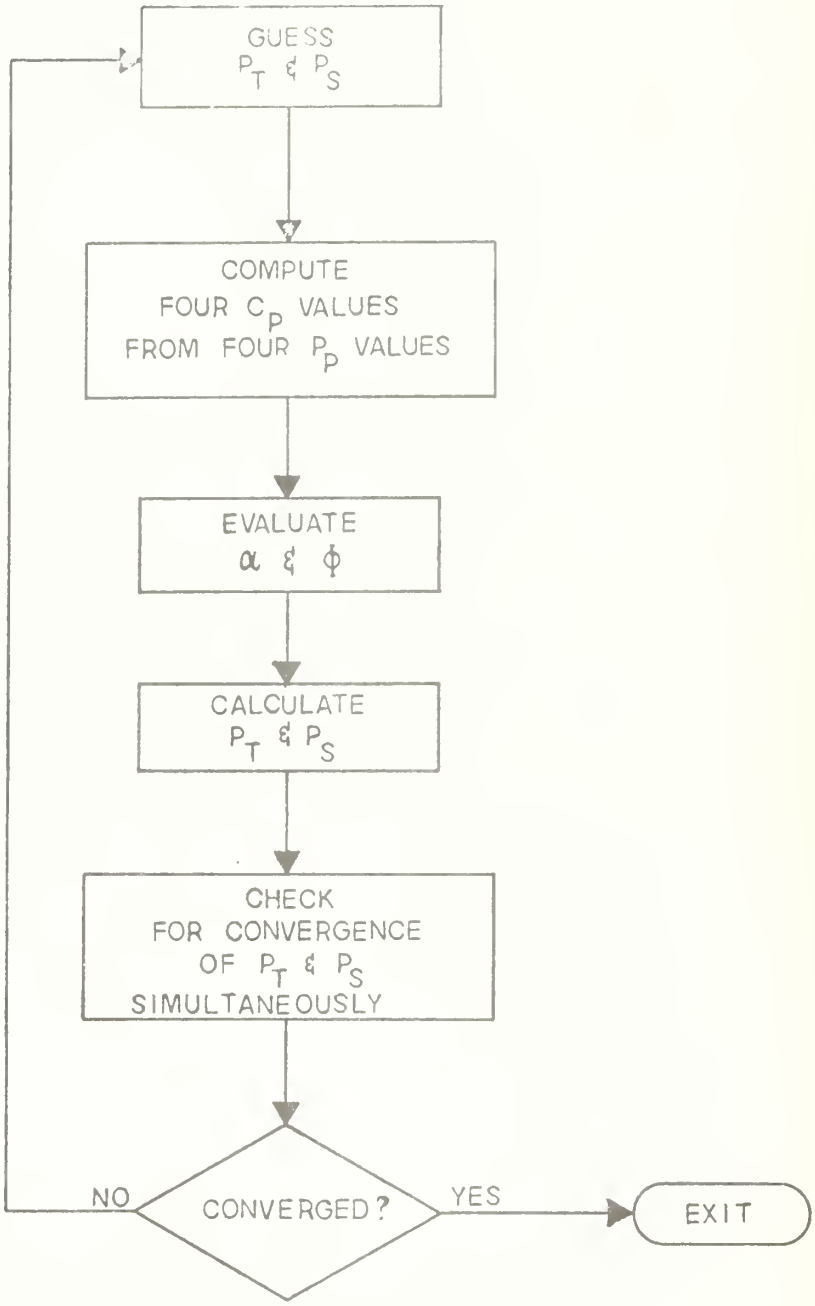


Fig. 2: Unstable iterative solution using measurement of A type and B type probes only.

fixed type B probe. The method of solution is then as shown in Fig. 3. The method shown in Fig. 3 proved to converge under all conditions. It is described in detail in the following pages.

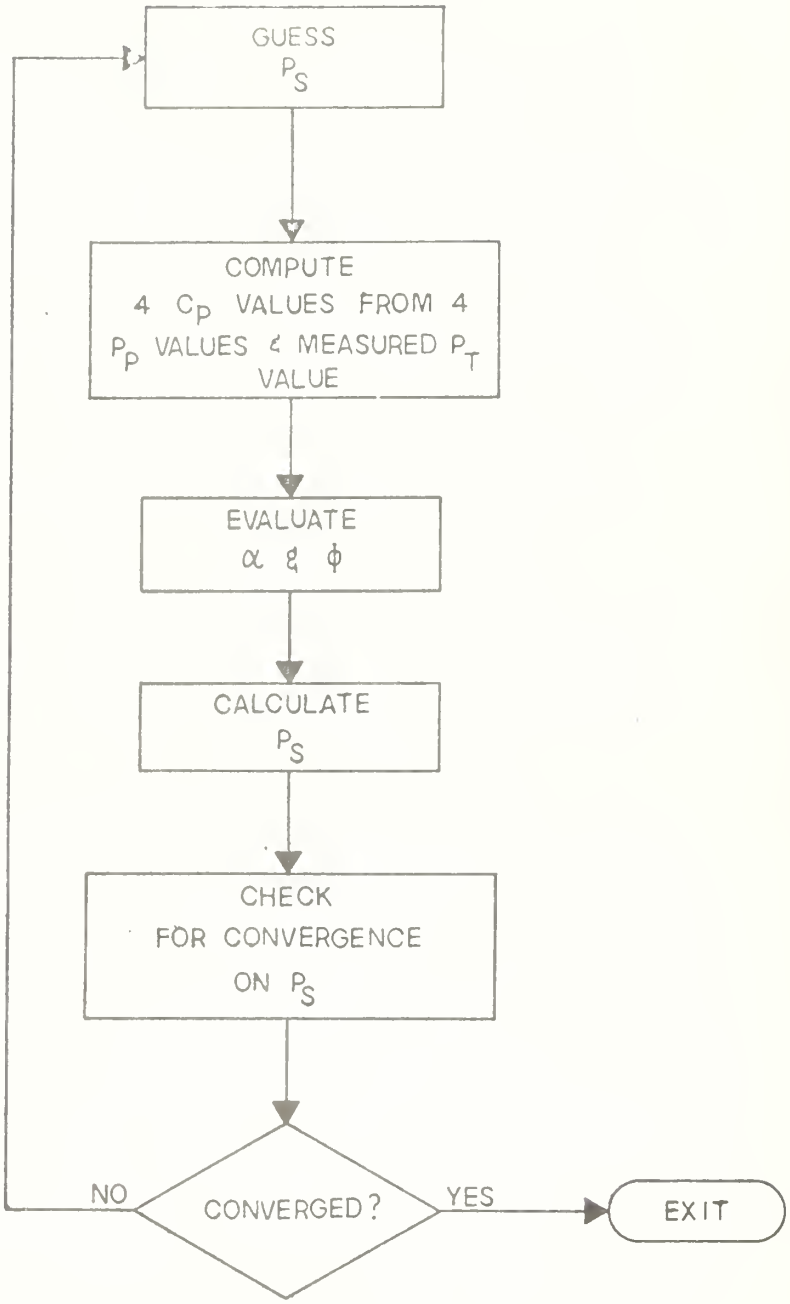


Fig. 3: Stable iterative solution using measurements of A type and B type probes as well as measurements of a Kiel probe.

Evaluation of α and ϕ (method)

The best way to understand the evaluation of yaw and pitch is to look at it from a topographical point of view. Each of the type A and type B probes has a unique calibration surface $Cp_{rp} = Cp_{rp}(\alpha_{rp}, \phi_{rp})$, where α_{rp} and ϕ_{rp} are measured relative to the axis of symmetry through the sensor at the probe tip. The calibration surface in this representation is invariant to yaw and pitch of the probe axis relative to the laboratory space. The same calibration surface if represented in the form $Cp = Cp(\alpha, \phi)$ can be written as

$$Cp = Cp[(\alpha_p + \alpha_{rp}), (\phi_p + \phi_{rp})] \tag{5}$$

where α_p and ϕ_p are the probe tip axis angular settings relative to the laboratory space. It is clear from eq. (5) that Cp can be derived from Cp_{rp} by a constant translation: α_p, ϕ_p on the α, ϕ plane. As each Cp_{rp} can be viewed as a hill with its peak at $\alpha_{rp} = 0$ and $\phi_{rp} = 0$. The Cp surfaces are the same hills with their peaks translated to $\alpha = \alpha_p$ and $\phi = \phi_p$.

In the present method probe A is used in three different angular settings namely:

I) $\alpha_p = 0 \quad \phi_p = 0$

III) $\alpha_p = \alpha_{pIII}$ $\phi_p = 0$

IV) $\alpha_p = \alpha_{pIV}$ $\phi_p = 0$

and the B probe is used in a fixed position, namely:

II) $\alpha_p = 0$ $\phi_p = \phi_{pII}$

For this case the topography of the calibration surfaces will appear as four hills. The three with their peaks at points (0,0); ($+\alpha_{pIII}$, 0) and ($-\alpha_{pIV}$, 0) are the translated hills Cp_{rpA} and the fourth with its peak at (0, ϕ_{pII}) is the translated hill Cp_{rpB}. Their contours of constant Cp are then as shown in figure 4.

Assume now that a velocity vector with yaw α and pitch ϕ is to be measured. These values of α and ϕ will be sensed uniquely by the probes at their four angular settings. Were equation (4) single valued, the values of α and ϕ could be uniquely evaluated, as the single intersection point between the projections of appropriate lines of constant Cp on the α, ϕ plane*. In the present case of the double valued functions, the lines of constant Cp are closed curves and more than a single intersection point do exist.

---

* Cp has generally a different value for each probe in each of its settings. Thus the solution involves solving for the intersection points between projections of contours of specified (but different) Cp values on the different hills.

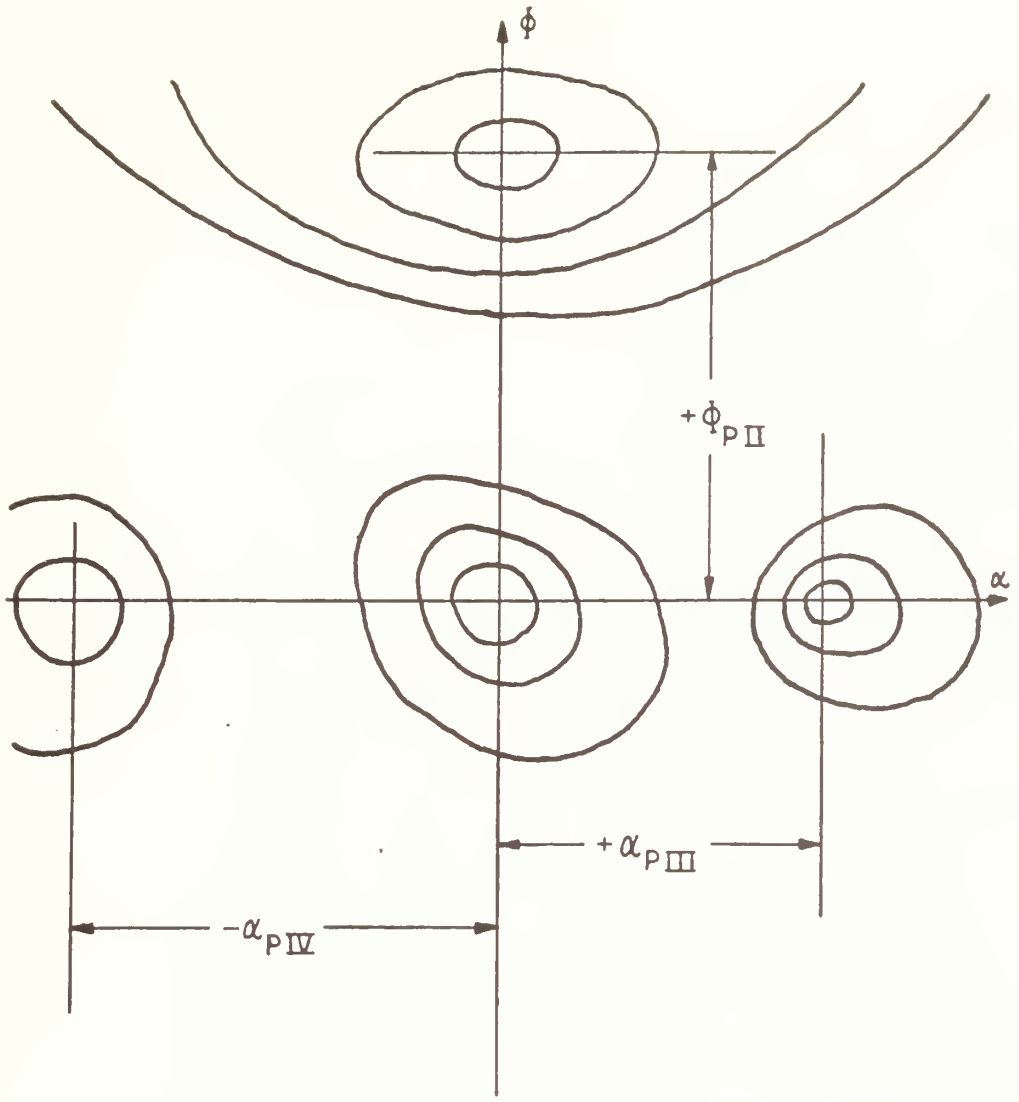


Fig. 4: Projections of C_p = const. lines of the four calibration surfaces (the center hill is for probe A at $\alpha_p = 0$, $\phi_p = 0$, the top hill is for probe B at $\alpha_p = 0$, $\phi_p = \phi_{pII}$, the right hill is for probe A at $\alpha_p = +\alpha_{pIII}$, $\phi_p = 0$ and the left hill is for probe A at $\alpha_p = \alpha_{pIV}$, $\phi_p = 0$.

The situation is shown in Fig. 5 for an example of a number of such intersection points. The correct intersection point, however, is uniquely identified as the only point through which all four Cp = const. curves pass.

From an examination of Fig. 5 it is clear that the intersection points of three closed Cp curves projections only are sufficient to identify α and ϕ uniquely. However one of these three curves must be that belonging to the type B probe.

The details of the numerical procedure used to obtain the correct intersection point in the evaluation of α and ϕ, are given in the following paragraph.

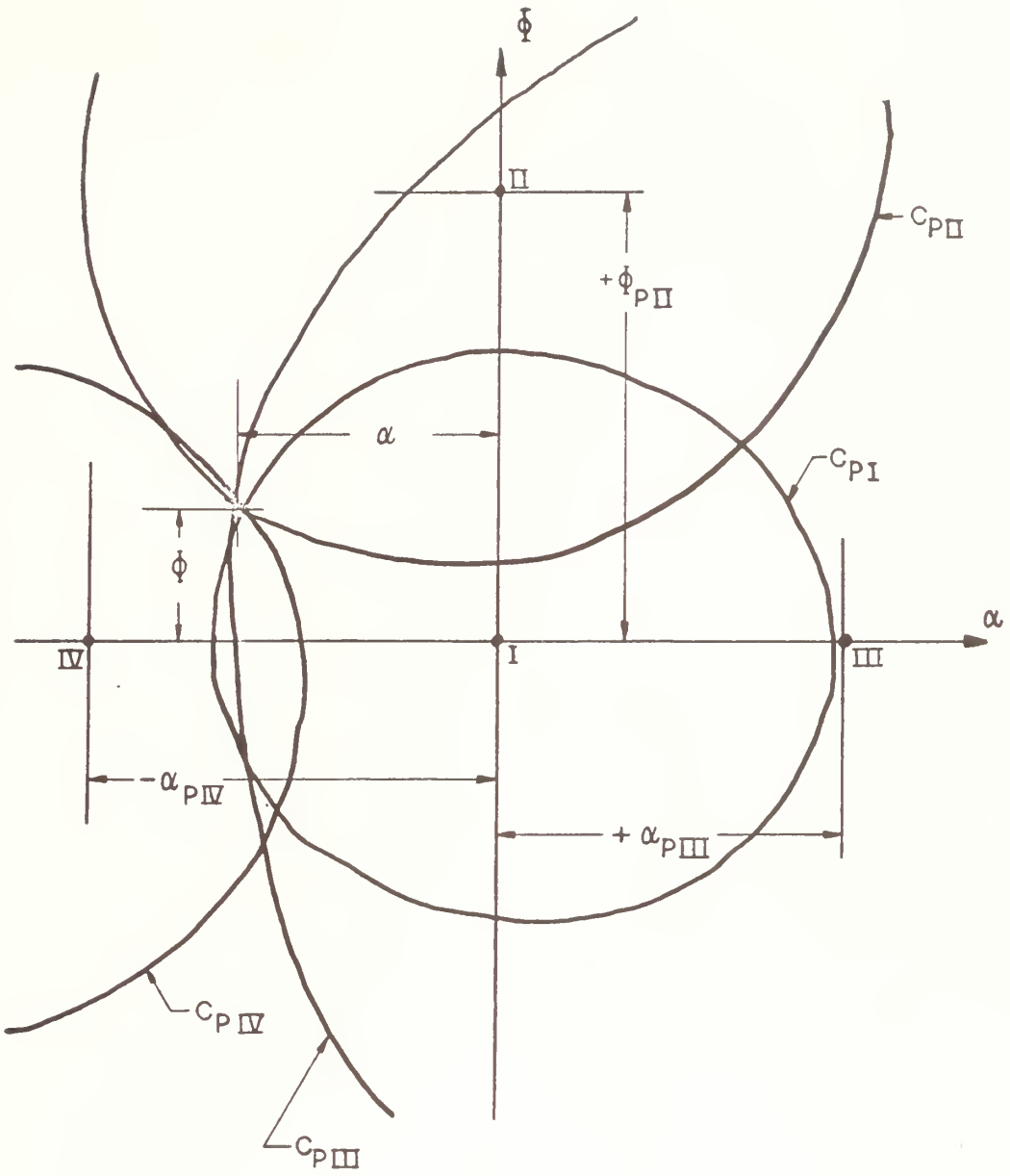


Fig. 5: An example showing a number of intersection points. The real one is in the second quadrant.

Evaluation of the intersection point coordinates (numerical Procedure)

The calibration surfaces Cp_{rp} are represented in form of a linear string of values ordered into an array, as shown schematically in figure 6. The numbers in Fig. 6 indicate the position of a Cp_{rp} value in the string. The string starts with the value of Cp_{rp} at a point $(-\alpha_{rpLo}; -\phi_{rpLo})$ and ends with the value of Cp_{rp} at a point $(+\alpha_{rpUp}; +\phi_{rpUp})$. This sequence must be kept and can not be changed.

Each of the closed Cp = const. curves projections in figs. 2 and 3 is the projection of the line of intersection between a plane parallel to the α,ϕ plane at a height equal to Cp and the calibration surface. These closed curves can be determined as the locus of the projections of all the penetration points of arbitrary lines parallel to the α,ϕ plane at a height Cp and the calibration surface. Such penetration points are calculated in subroutine "PENPTS", Fig. 10.

PENPTS calculates the first two penetration points of a surface Z (in the present case Z = Cp) by a straight line piercing that surface. If the surface is double valued these two points are the only roots. The surface is given as a table of numbers on a cartesian basis Z = Z(X,Y) (or in the present case $Cp = Cp(\alpha,\phi)$.

The subroutine has the following limitations:

1) No roots can be found on the lower Y = const. boundary

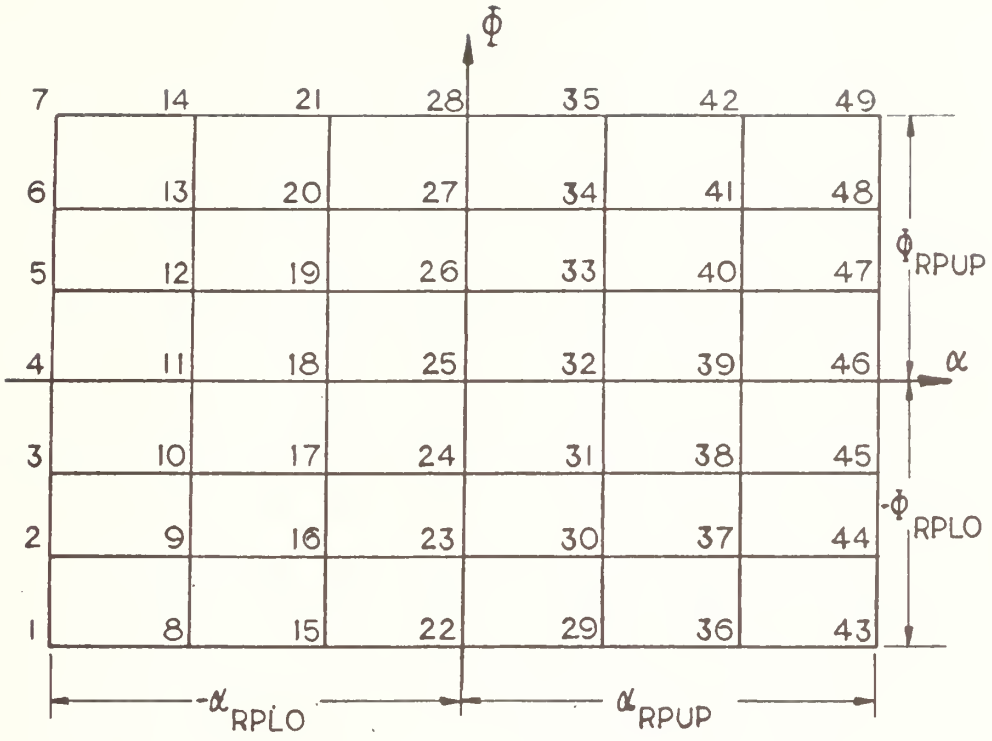


Fig. 6: The order of a calibration surface array.

2) The roots can be found only along a piercing line which is parallel to the X axis (but at any height above the X,Y plane)

3) The table Z = Z(X,Y) defining the surface must be based on a grid comprising lines X = const. and Y = const. The spacing of these lines must not be equal. In other words the surface is defined by a rectangular grid in the X,Y plane, from X_{min} to X_{max} and from Y_{min} to Y_{max}.

4) Only a single root can be evaluated in a surface element located above a defining rectangle on the X,Y plane.

5) Not more than the first two roots will be evaluated for any piercing line.

6) The surface must be monotonic over each rectangle (this is a result of limitation 4).

7) All X and Y arguments must be given in increasing order.

These limitations do not restrict the application of PENPTS in the present problem as long as the calibration surfaces are smooth within the element located above a grid rectangle. However, the elemental grid rectangles can be reduced arbitrarily in size. If a calibration surface is more than double valued PENPTS will fail. However in this case the probe yielding such a calibration surface can not be considered a useful instrument unless used only in parts of the domain where it is double valued.

Z is defined from X_{min} to X_{max} and from Y_{min} to Y_{max}. PENPTS is given the following initial information: X values and Y values defining grid points, corresponding Z values, the Y location (YG) and the height above the X,Y plane (HT) of the piercing line. With this information PENPTS searches for the band of rectangles which includes YG or of which YG is the lower boundary (see fig. 7) and then scans this band from left to right in search for penetration points. The scanning is based on the geometry given in fig. 7 which represents a particular element in the band, approximated by two plane triangles.

Initially a coarse scan is carried out just to detect, but not to evaluate, an intersection point. This is done checking for each sub-domain whether (ZX(I-1) - HT)/(ZX(I) - HT) 0. If this condition is true a root is detected and control is tranferred to its exact evaluation. The value of the root is calculated after its location, either in the first (left) or second (right) triangle is determined (each grid rectangle is composed of two triangles). Equation 6 which is based on fig. 8 (for a left triangle) or eq. 7 which is based on fig. 9 (for a right triangle) is used to evaluate the penetration point. These equations express the linear interpolation of Cp in the Fortran rotation used in this program.

```
XS = X(I-1) + ABS(ZX(I-1)-HT)*(XM-X(I-1))/(ABS(HT-ZM) +
   + ABS(ZX(I-1)-HT))                                    (6)
```

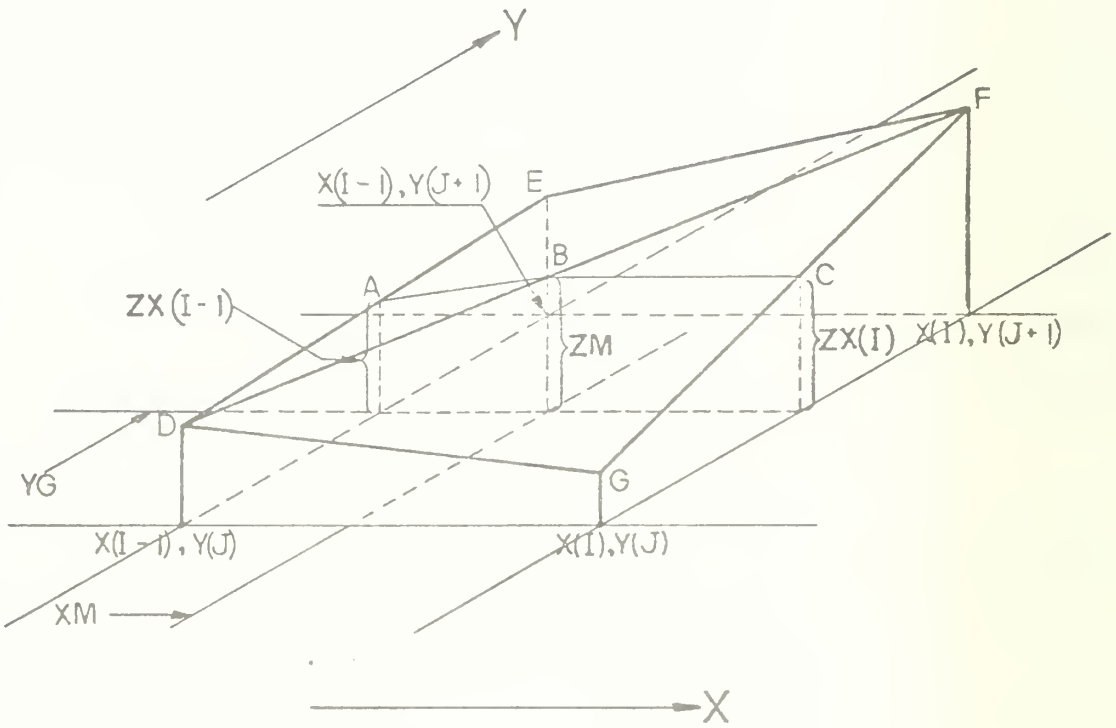


Fig. 7: The geometry of a linearized calibration surface element comprising two plane triangles and its intersection with a plane normal to X,Y along Y = YG.

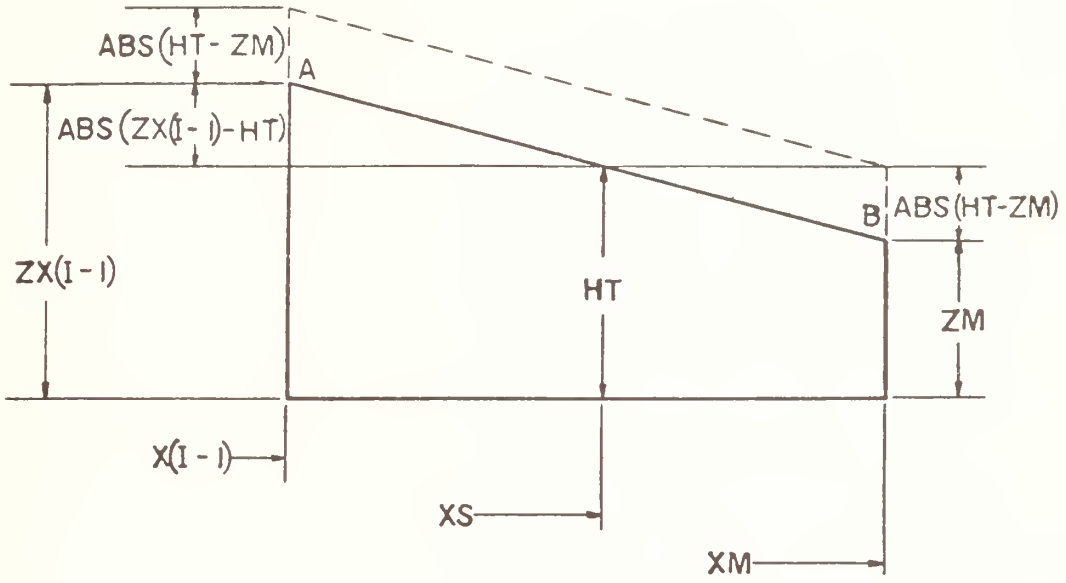


Fig. 8: The geometry for a penetration through a left hand (first) triangle.

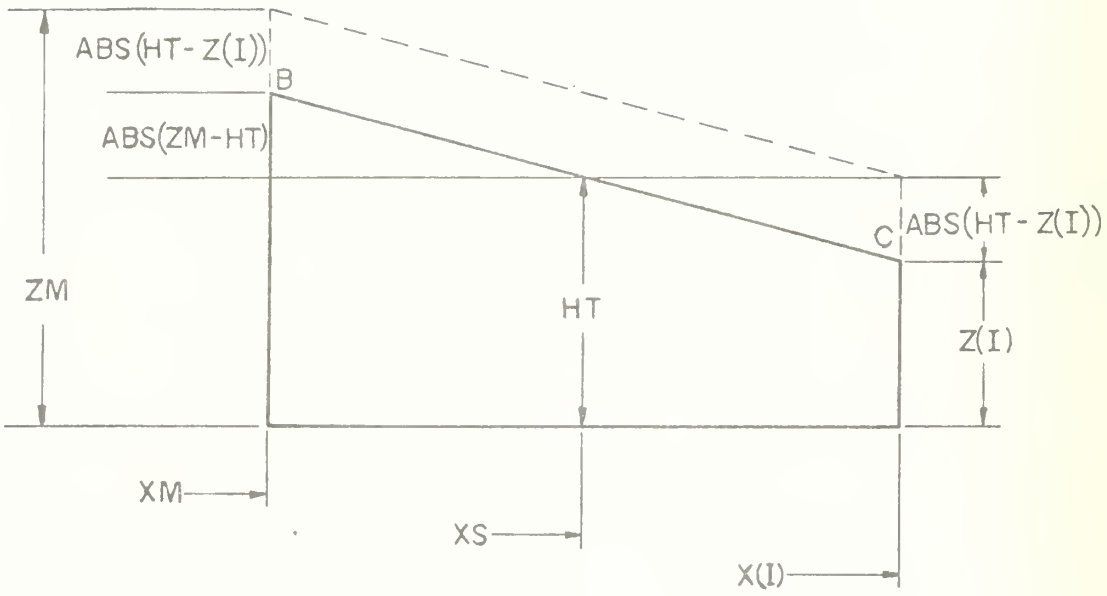


Fig. 9: The geometry for a penetration through a right hand (second) triangle.

```
XS = XM + (X(I)-XM)*ABS(ZM-HT)/(ABS(ZM-HT) +
     ABS(HT-Z(I)))                                    (7)
```

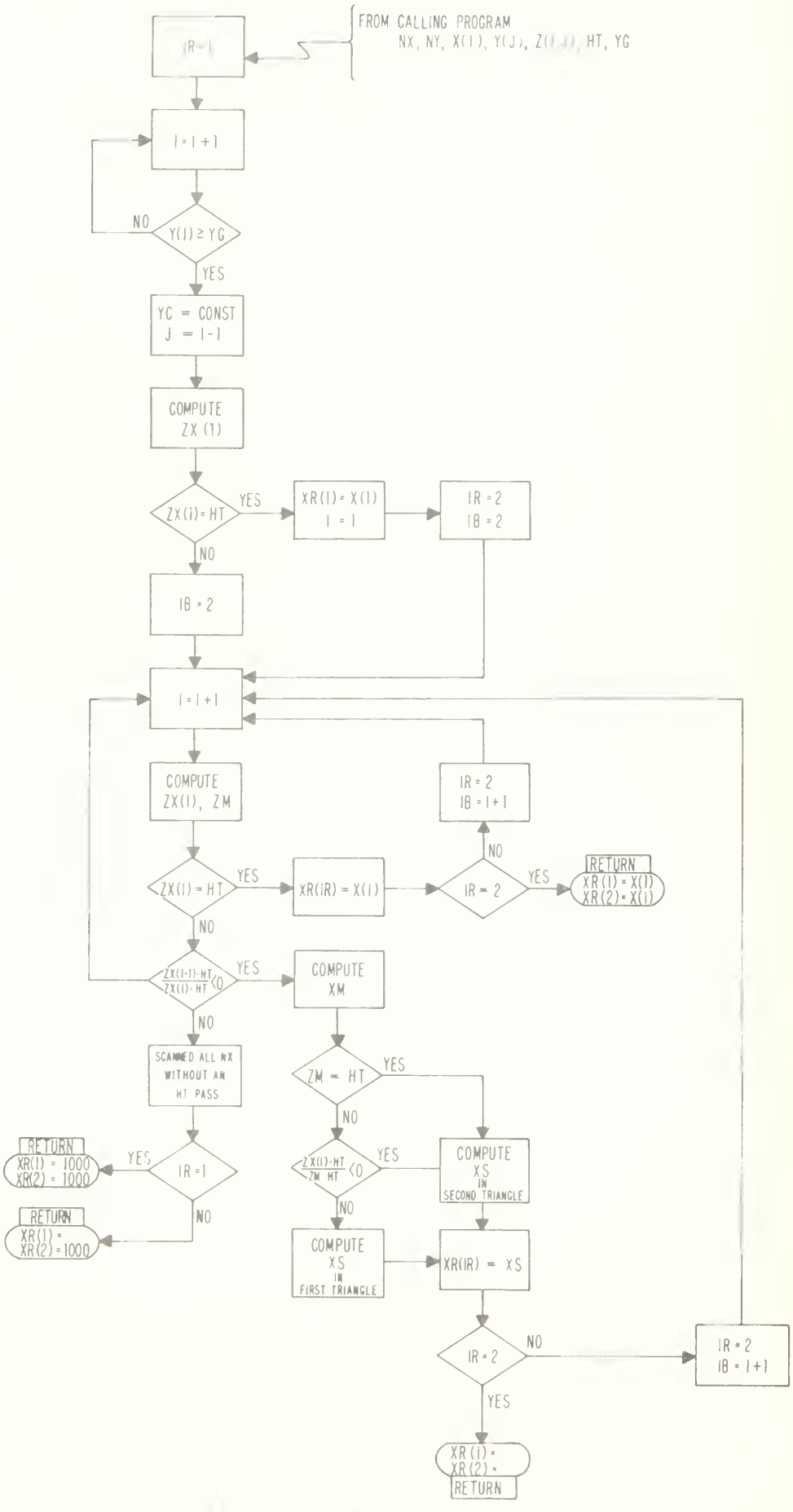
Equations 6 and 7 are invariant to the slope of the calibration surface i.e., the slope of the straight lines AB and BC.

Finally it should be pointed out that when two penetration points are determined the values of their abscissas, X(I), are returned. When only a single penetration point is detected the abscissa X(2) will be returned with a value of 1000.0. When no penetration point is determined both X(I) values returned will have the value of 1000.0. The program logic is designed to recognize these messages.

It was stated earlier that PENPTS is used to determine the closed intersection curves projections on the X,Y plane. In fact not the curves but just the intersection points between each two of them are required (see figure 3).

To compute the coordinates of these points the subroutine "INTSCS" is used. It uses PENPTS as a subroutine.

In INTSCS a scanning procedure is carried out from a minimal value of Y (or ϕ) YRIN to an upper value of YRUP, or a


FROM CALLING PROGRAM
NX, NY, X(I), Y(J), Z(I,J), HT, YG
IR=1
I=I+1
NO
Y(I)≥YG
YES
YC = CONST
J = I-1
COMPUTE
ZX (1)
ZX(i)=HT
YES
XR(1)=X(1)
I = 1
IR = 2
IB = 2
NO
IB = 2
I=I+1
COMPUTE
ZX(I), ZM
IR = 2
IB = I+1
NO
ZX(I)=HT
YES
XR(IR)=X(I)
IR = 2
YES
RETURN
XR(1)=X(1)
XR(2)=X(I)
NO
ZX(I-1)-HT / ZX(I)-HT <0
YES
COMPUTE
XM
NO
SCANNED ALL NX
WITHOUT AN
HT PASS
ZM = HT
YES
NO
RETURN
XR(1) = 1000
XR(2) = 1000
YES
IR=1
NO
RETURN
XR(1) =
XR(2) = 1000
ZX(I)-HT / ZM HT <0
YES
COMPUTE
XS
IN
SECOND TRIANGLE
NO
COMPUTE
XS
IN
FIRST TRIANGLE
XR(IR) = XS
IR = 2
NO
IR = 2
IB = I+1
YES
XR(1) =
XR(2) =
RETURN

prescribed 10,000 times* which ever comes earlier. The subroutine scans through any two arbitrarily chosen closed curve projections to find their intersection points. In each scan (I) up to four penetration points can be determined, while the penetration points of the previous scan (J) are memorized. Together, eight penetration points can be involved. When no intersection point exists the geometrical situation is as shown in figure 11, while the existance of an intersection point is characterized in figure 12. Subroutine INTSCS can distinguish between the two situations. In figures 11 and 12 the case of four penetration points found in each scan are shown. The subroutine, however will handle any possible number of such points, from zero to four. A "no penetration points" is assigned an abscissa value of 1000.0 by PENPTS as explained earlier.

When an intersection point is detected its evaluation is based on the geometry shown in fig. 12. The eight penetration points have the following coordinates:

AJL(RAJL,YRJ)	BJL(RBJL,YRJ)
BJR(RBJR,YRJ)	AJR(RAJR,YRJ)
BIL(RBIL,YR)	AIL(RAIL,YR)
AIR(RAIR,YR)	BIR(RBIR,YR)

---

* The 10,000 scans are governed by a program constant in the line "DO 140 ...." and can be arbitrarily varied.

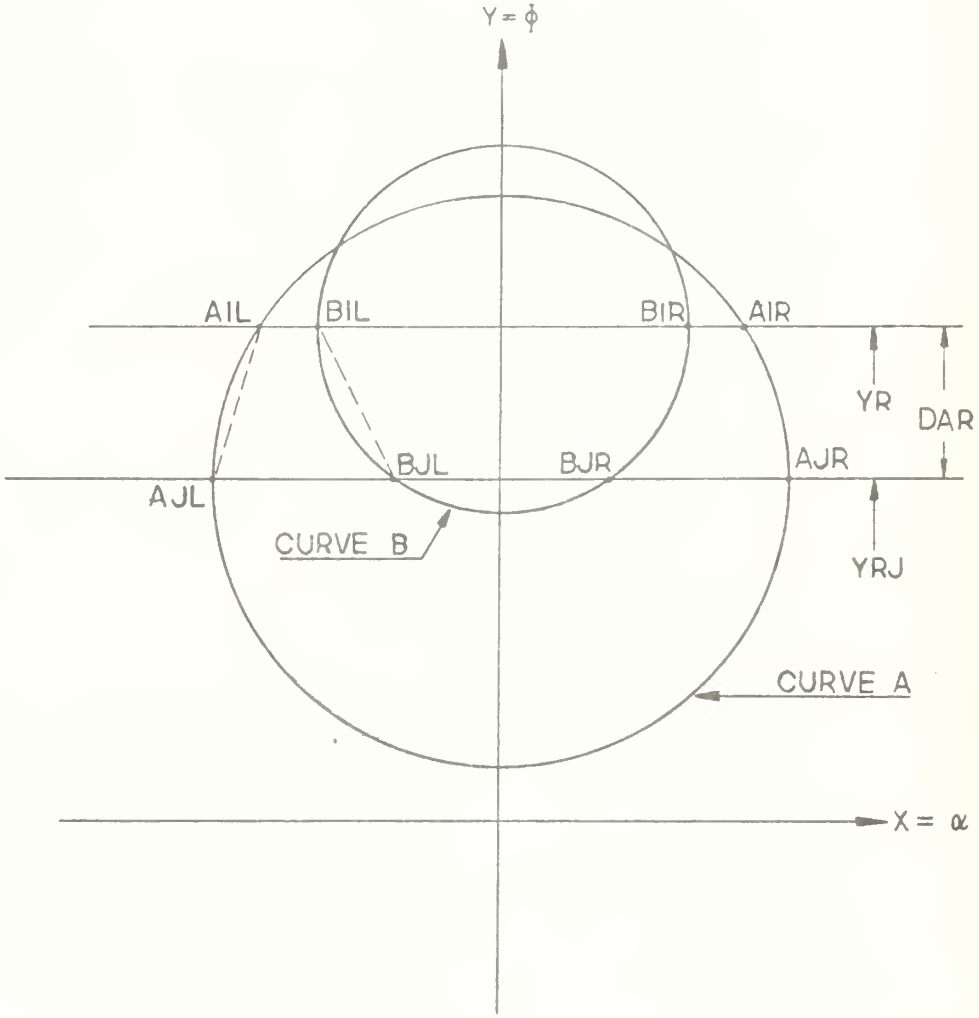


Fig. 11: The geometry for two successive scans when no intersection point exists.

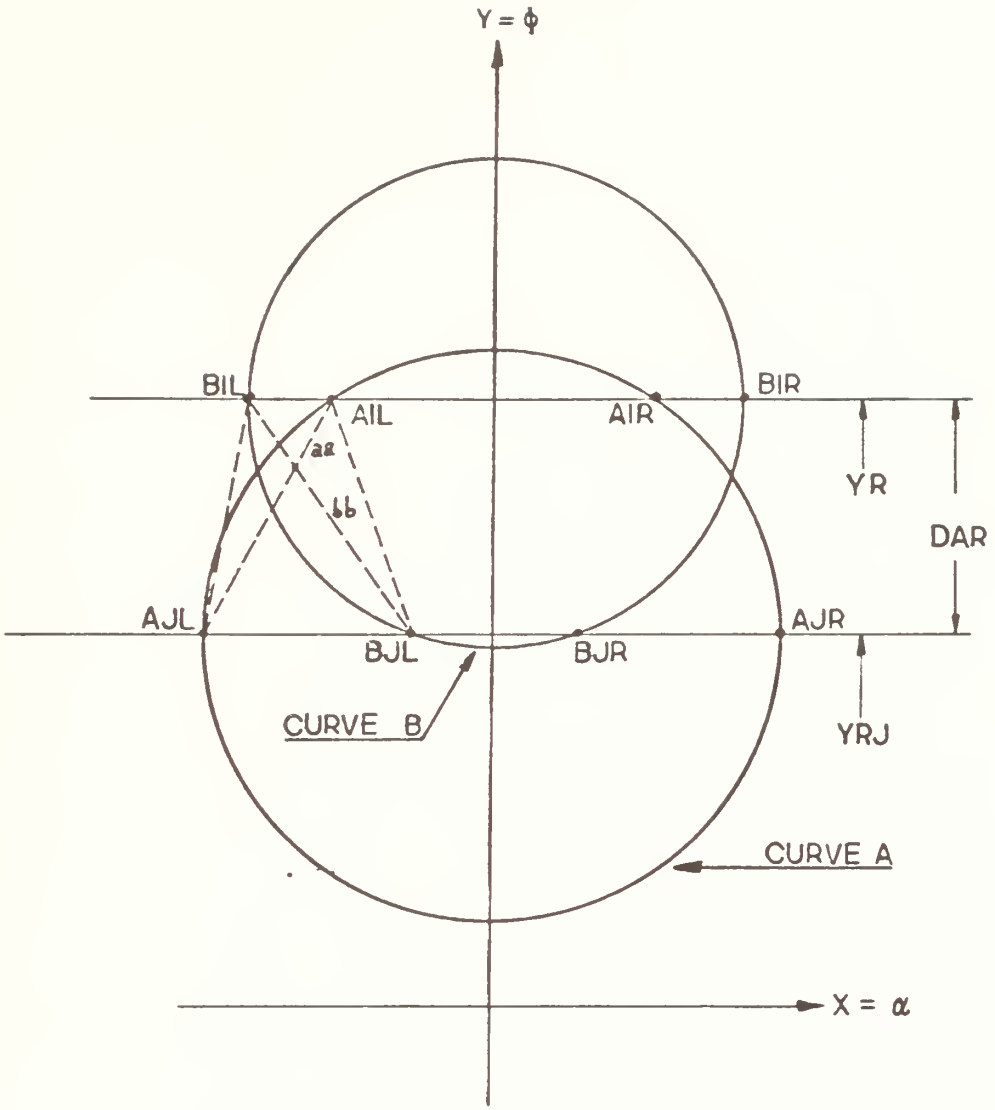


Fig. 12: The geometry for two successive scans when an intersection point does exist.

The left intersection point illustrated in fig. 12 is the intersection point of the lines aa and bb, each described by its equation:

for aa $Y = (AA)X + BA$ (8)

for bb $Y = (AB)X + BB$ (9)

The constants AA, BA, AB and BB are given in equations 10 to 13.

$AA = (YRJ-YR)/(RAJL - RAIL)$ (10)

$BA = YR - AA*RAIL$ (11)

$AB = (YRJ-YR)/(RBJL-RBIL)$ (12)

$BB = YR - AB*RBIL$ (13)

The coordinates of the intersection point to be calculated are

$X = (BB-BA)/(AA-AB)$ (14)

$Y = X*AA + BA$ (15)

These relations are true for a left hand intersection point. Analogous equations are true for a right hand intersection point. In this algorithm the straight lines aa and bb approximate the curved lines connecting AJL with AIL and BJL with BIL or similar lines on the right hand side of figure 12. The error introduced through this approximation is reduced as DAR = $\Delta\phi$ is reduced.

It is possible that an intersection point is identical with AIL and BIL or AIR and BIR. This case is defined as "direct hit". The program is designed to detect such a direct hit and evaluate the corresponding intersection point accordingly.

The above algorithm works perfectly as long as the two closed Cp = const. curves are far from being tangent. But in

practice a situation of almost tangent curves can arise when X (or α) is very small. In this case the preceeding algorithm will fail and must be replaced. The geometry of this situation is described in figure 13. This situation is identified by INTSCS and the intersection points are then evaluated assuming that they are intersections of two circular arcs. When the curves Cp = const. are almost circular this approximation does not lead to unacceptable errors.

It was stated earlier that INTSCS scans from a minimum value of ϕ to a maximum value of ϕ with prescribed steps $\Delta\phi$, as shown in figs. 11 and 12. This direction of the scanning is used when the intersection points of the curves Cp_I and Cp_{II} of fig. 5 are evaluated.

However, in course of the reduction of the measured data, scans in the direction of α in steps of $\Delta\alpha$ are also necessary to evaluate the intersections of the curves Cp_I and Cp_{III} or Cp_I and Cp_{IV}. INTSCS is designed to carry out this task as well. To do this the calling statement for INTSCS is appropriately changed as will be explained in the next section. To be general enough INTSCS is not written in terms of α and ϕ or X and Y but rather in terms of general arguments. The best way to understand INTSCS is to compare its general arguments to physical quantities by means of the calling statements. In fig. 14 the flow diagram in INTSCS is given in terms of the general arguments.

INTSCS returns to the main program (fig. 15) the (α,ϕ) coordinates of two intersection points between the Cp = const. curves specified in the calling statement. Let us now follow the way in which the main program is designed to utilize INTSCS

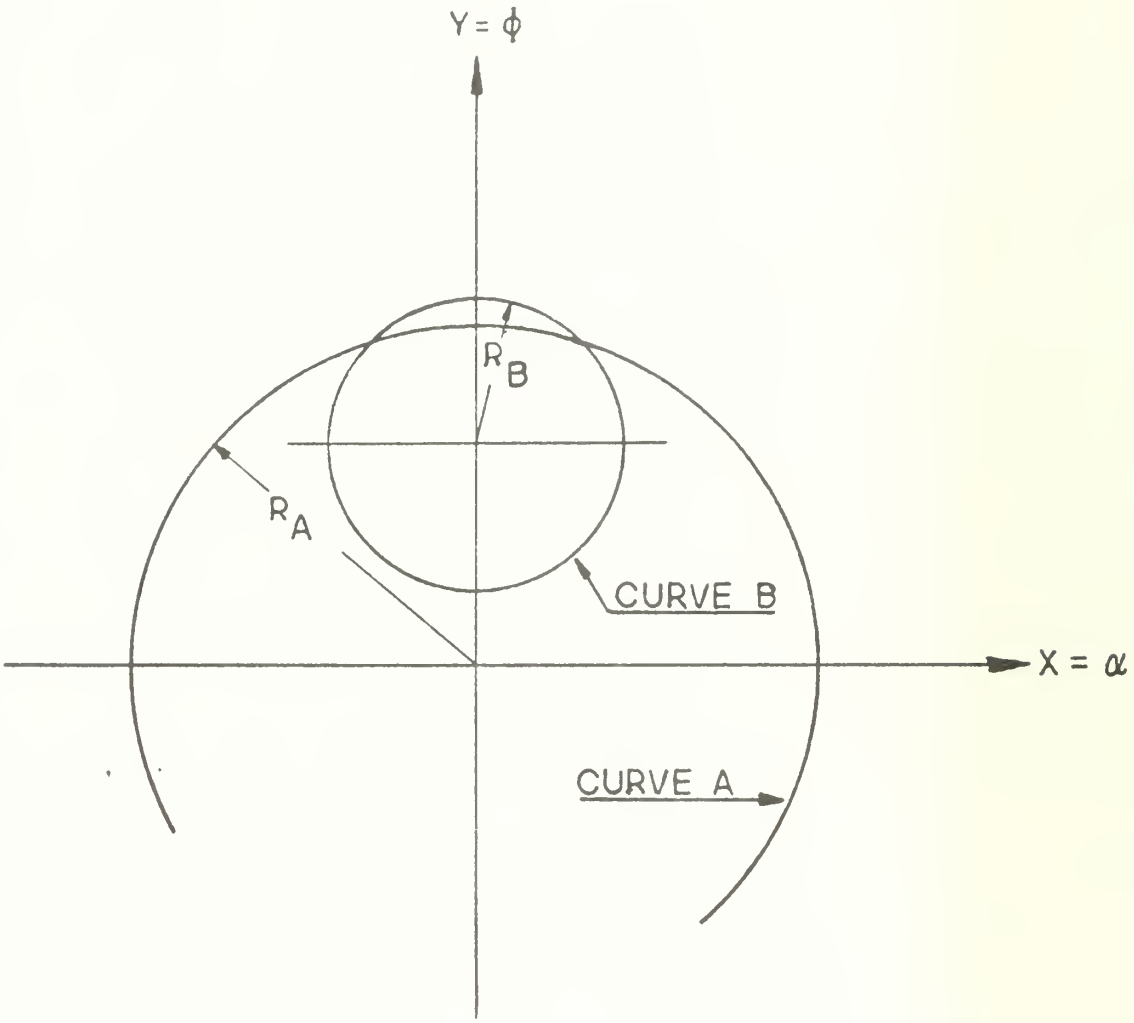


Fig. 13: The geometry when two closed C_p = const. curves are almost tangent.

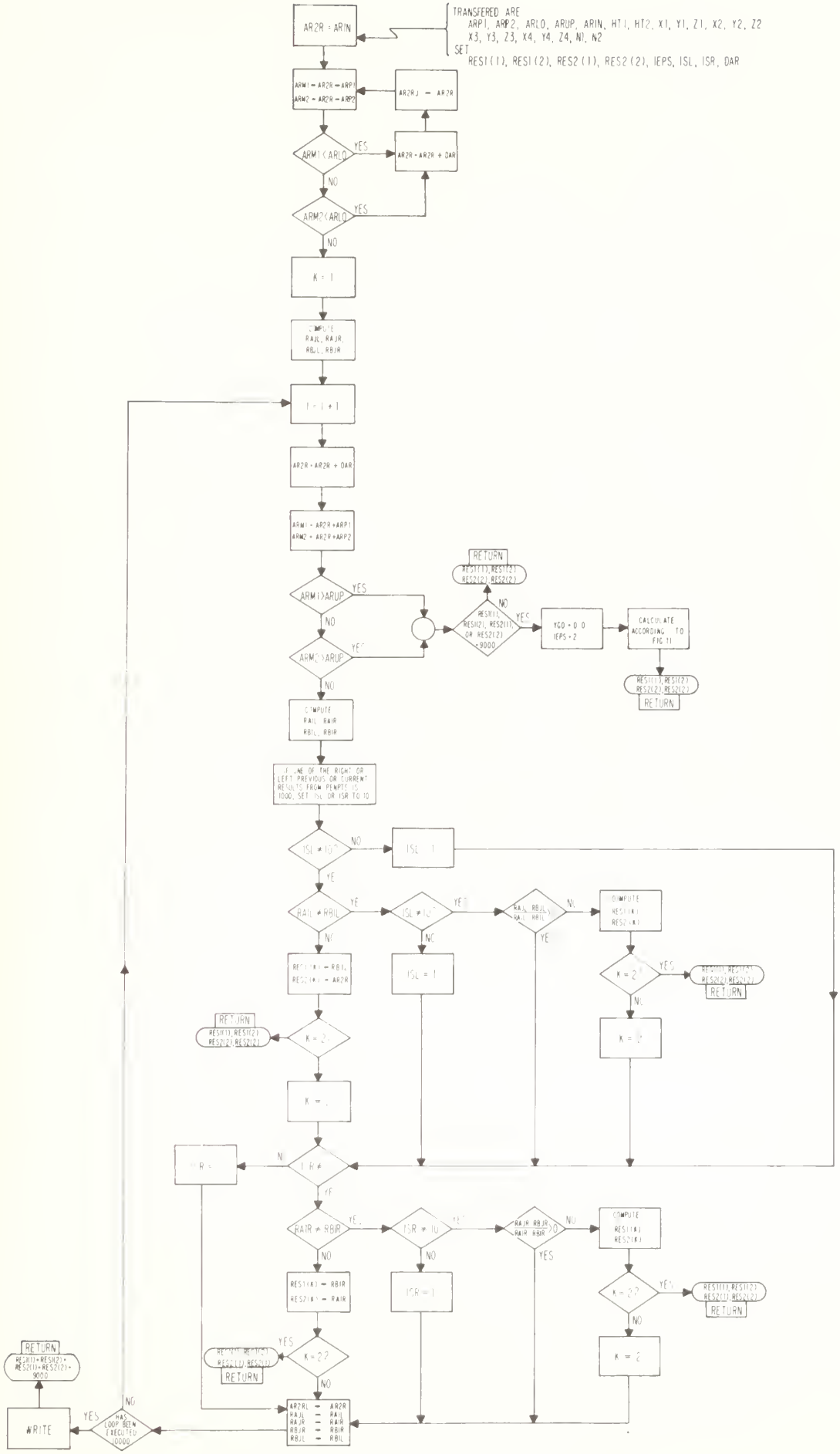


FIGURE 14. FLOW DIAGRAM OF INTSCS

for the evaluation of α and ϕ of the velocity vector, as well as static and total pressures.

Evaluation of the Velocity Vector and Pressures from the Probe Signals

Fortran program VDR was written to evaluate the velocity vector from probe measurements of pressures. The program is shown in Fig. 15.

At line 1410 INTSCS is called to scan curves I and II for possible intersection points. Scanning can be carried out in the direction of the ordinate only, with the calibration curve matrices compiled exactly as shown in fig. 6. This limitation is imposed by the way PENPTS is constructed. In the case of the intersection points between I and II ZA and ZB are scanned without difficulties in the direction of the ordinate which is ϕ as shown in fig. 12 and returns with the coordinates of the two first intersection points, points 1 and 2 of Fig. 16. They are ALF1, PHI1 and ALF2, PHI2. In line 1500 INTSCS is called again to scan curves I and III. Now scanning has to be carried out in the direction of the abscissa, a task for which INTSCS was not designed. To overcome this problem the calibration curve matrices are used in a transformed form such that the previous abscissas are now ordinates, ordinates are abscissas and the internal structures of the Cp_{rp} arrays are accordingly modified. This transformation is carried in VDR in the section between lines 320 and 450. Comparison of the calling line 1500 to the previous calling line 1410 shows very clearly how the various arrays: original and transformed, are used. The coordinates of points 3 and 4 of fig. 16

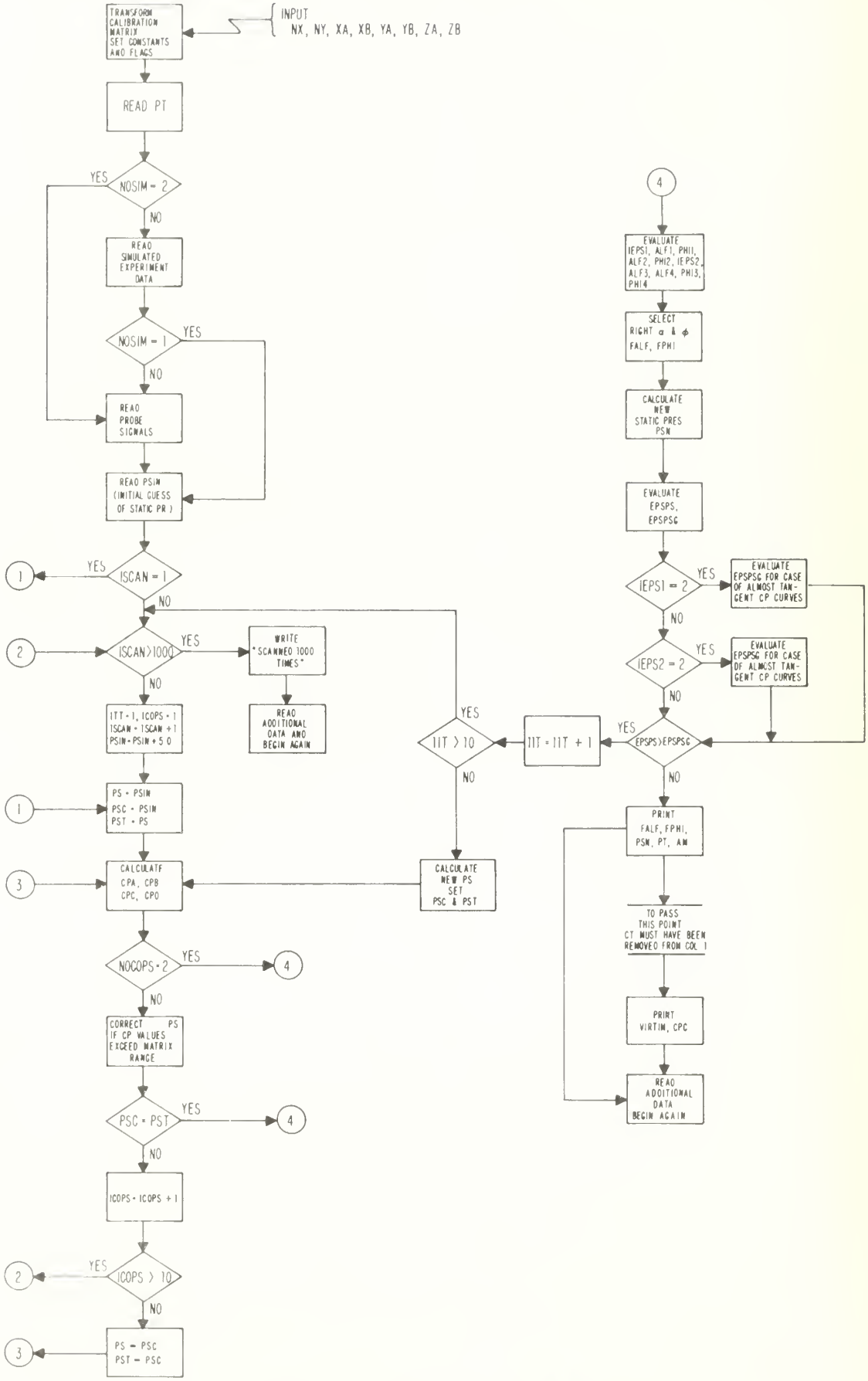


FIGURE 15. FLOW DIAGRAM OF VDR

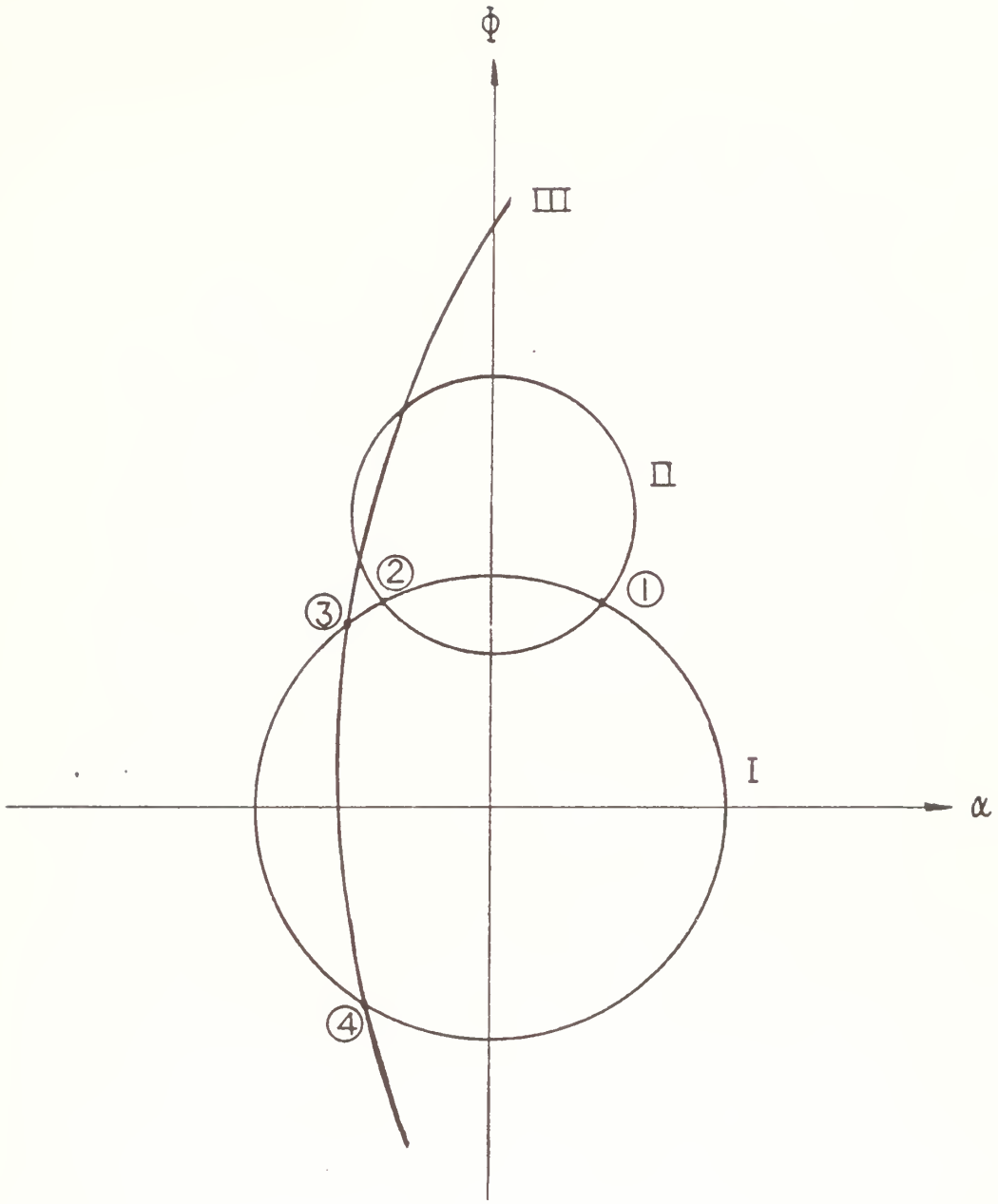


Fig. 16: Selection of the proper intersection point in VDR.

are now returned to the main program of VDR, they are ALF3,PHI3 and ALF4,PHI4.

We are seeking a single intersection point; the one representing yaw and pitch as sensed by probe A in positions I and III and probe B in position II. But, as the calibration surfaces are double valued we are now, unfortunately, in the possession of four points. The solution, however, is physically unique. Only a single velocity vector exists in reality and its yaw and pitch are included in the four intersection points evaluated. Were both the measurements and the numerical procedure absolutely accurate, two of the four points would have been identical. But this is not the case in reality, instead of a single point, two points close to each other will be detected. Therefore the average coordinates of the two of the four intersection points returned to VDR from INTSCS which are closest to each other are selected as the measured yaw and pitch angles. In the example shown in fig. 16 this will be the point dividing the distance between points 2 and 3.

The calculation is now at the point at which α and ϕ are temporarily known (see fig. 3). Using α and ϕ a new Cp_{rpIV} (for probe A in position IV) is computed by linear interpolation using subroutine INTPLT. With this new value of Cp_{rpIV} and with Pp_{IV} a new P_s is computed. If this new value of P_s is close enough as determined by EPSPSG to the guessed value, or to the value of P_s in the previous iteration, the data reduction for the particular point in question is terminated.

The relative difference between present and previous P_s is compared to the convergence criterion EPSPSG. This criterion is evaluated by an empirical function determined to give best compromise between accuracy of results and computer time required until convergence is achieved. When the calculation follows normal routine the convergence criterion is given in line 1710 as function of P_T . When the routine for almost tangent curves is used during data reduction either on the right or the left side a different empirical function (lines 1720 or 1730) is used.

If the convergence criterion is satisfied results are printed out and data for a new measurement point is read for reduction. If, however convergence is not reached a new static pressure for a next iteration is evaluated (line 1780).

Convergence and accuracy

The convergence of the present iterative procedure is not ensured in all possible cases of data sets. The program, however, is adjusted to converge in most of the cases. Similarly to other iterative computation method some experience is required to achieve convergence when the calculation does not converge. In this paragraph the principal factors affecting convergence, computation time and accuracy, which are abviously coupled, are pointed out.

The convergence and accuracy of reduction of a data set: P_T, PPA, PPB, PPC, PPD, PSIN (or analogous set in the experiment simulation mode) depends on the following factors, which can be varied by the user.

1) Coarseness of calibration arrays ZA, ZB and their linearity. With coarser arrays convergence problem will increase and accuracy of results decrease.

2) Probe settings YPB, XPC, XPD (with the following probe settings being fixed and not variable YPA = YPC = YPD = 0.0; XPA = XPB = 0.0). The optimum setting is about $\pm 20^{o}$ to 25^{o}. Too small values reduce accuracy, too big values cause convergence problems and probe tip flow separation.

3) RELXPS, the static pressure relaxation factor. The smaller this factor the safer will convergence be achieved. Computation time, however, will increase.

4) The constant 5.0 in line 1050 of the program. The dimension of this constant is kg/m^2. If too big results can be lost and if too small computer time will be growing. The variation of this constant should be coupled with an appropriate modification of the constant in line 1010 of VDR (item 8 in this list).

5) The constants in the evaluation of EPSPSG (lines 1710 to 1730). The smaller EPSPSG the more accuarte the results in expense of increased computation time and reduced convergence safety.

6) The constant scanning step DAR. The smaller DAR the more accurate the results, but too small values can cause complete loss of results. Computation time increases with reduced DAR.

7) The constant 10 in line 1360. This constant governs the number of P_s corrections.

8) The constant 1000 in line 1010. This constant governs the number of scans.

If convergence is not achieved in a particular case variation of each or of a combination of the above values will always enable convergence.

Evaluation of the Computation Time

By deleting the letters CT from column 1 and 2 of lines 20 to 70, 1930 to 1960, 1980, 2000, 2010 the actual computation time as well as CPU time are evaluated and printed out. The following statement prior to execution is required in this case:*

GLOBAL T SYSLIB SSPLIB

This option is useful for adjusting the constants affecting convergence as to optimal compromise between accuracy, ease of convergence and calculation costs.

* When the program is run on Naval Postgraduate School IBM 360 system.

Input

1) Calibration arrays have to be input in the following manner: first values of NX and NNY , second values of X in rising order, third values of Y in rising order and fourth values of Z in the order shown in Fig. 6. Two calibration arrays are read: first the array of type A probe and second the array of type B probe.

2) If the program is run in experiment simulation mode PT and PS are read in Kg/m^2 and subsequently $\circ$ and ϕ .

3) If the program is run in data reduction mode the measured values of PT, PPA, PPB, PPC and PPD are read and then the guess of static pressure PSIN. All are read in Kg/m^2. PSIN has to be lower than the actually existing value to ensure convergence of the calculation. Too low a value will cause waste of computer time.

Calibration arrays are read from a disk space on which they are stored. The following statement prior to execution is required for successful reading:

FILEDEF 02 DSK NAME XX

here NAME is the name of the file on which the calibration data of probes A and B is stored in proper order and format, XX is a two digit number.

Simulation or measurements reduction input is read in the normal way using the terminal keyboard or punched cards.

Output

α, ϕ, P_S, P_T and Ma are printed out according to lines 1820 to 1920. This output, however is not sufficient when the logic of the computation is to be followed either to examine the execution of the program or for debugging. Two programs with additional output are given in appendices 2 and 3 and can be used for this purpose. The first, SWVDR gives short additional output, namely:

1) When PSIN is successively increased automatically by the program to ensure convergence, the values of PSIN are printed out.

2) IIT, FALF, FPHI and PSN are printed out at the end of each iteration prior to the convergence test.

The second program WVDR prints more detailed information. All additional WRITE statements in this program are numbered with three digit numbers starting with 9.

Conclusions

The following conclusions are drawn on the basis of experience gained in running the program with various data sets and various types of calibration surfaces.

1) Optimal probe settings for ease of convergence and high accuracy are: YPB = 25°, XPC = 25°, XPD = -25°. It is therefore suggested that a B type probe with 25° pitch will be used, and that the A type probe be rotated to $\pm$ 25°.

2) Convergence and accuracy, as well as computer time efficiency are improved when the calibration surfaces of both probes are not flat at their peaks but are rather rounded. It is therefore suggested that a new probe tip geometry be considered. A spherical tip with a central pressure tap is recommended. To prevent damage to the sensitive transducer located behind the pressure tap, and in order to improve the frequency response of the probes it is suggested that the volume between the pressure tap face and the transducer be filled with an appropriate liquid that will not affect the transducer negatively. In this case the opening of the pressure tap has to be sealed with a very thin low inertia membrane.

3) It is probably possible to modify the iterative procedure such that safe convergence can be achieved also when using the scheme of Fig. 2. If this can be achieved the Kiel probe will not be necessary. An effort in this direction is suggested.

References

1. Dunker R. J., Strinning P. E., and Weyer, H. B., "Experimental Study of the Flow Field Within a Transonic Axial Compressor Rotor by Laser Velocimetry and Comparison With Through-Flow Calculations", ASME Journal of Engineering for Power, Vol. 100, pp. 279-286, April 1978.

2. Shreeve, P. P., Simmons J. M., Winters K. A., and West J. C. Jr. "Determination of Transonic Compressor Flow Field by Synchronized Sampling of Stationary Fast Response Transducers", Symposium on Non-Steady Fluid Dynamics, ASME 1978 Winter Annual Meeting, San Francisco, Dec 1978. (To be published in ASME Journal of Fluids Engineering)

3. Thompkins W. T. Jr., and Kerrebrock J. L., "Exit Flow From a Transonic Compressor Rotor", AGARD Conference Proceedings No. 177, Unsteady Phenomena in Turbomachinery, pp. 6-1 to 6-23. Meeting held at Naval Postgraduate School, Monterey, California, 22-26 September 1975.

4. Shreeve R. P., McGuire A. G., and Hammer J. A., "Calibration of a Two Probe Synhronized Sampling Technique for Measuring Flows Behind Rotors", paper to be presented at IEEE, Eighth International Congress in Instrumentation in Aerospace Simulation Facilities, Naval Postgraduate School, Monterey, September 24-26 1979. Published as IEEE ICIASF Record of Proceedings.

APPENDIX I

LISTING OF VDR

07/08/79 18.43.34

FILE: VDR FORTRAN P1 N P S

```
C  START OF VDR                                                          INT00010
CT      INTEGER*4 A(6)                                                  INT00020
CT      IASC=0                                                          INT00030
CT      IAC0=C                                                          INT00040
        DIMENSION ALF(40),PHI(40),XA(40),YA(40),                        INT00050
       1ZA(40,40),XB(40),YB(40),ZB(40,40),XAX(40),YAX(40),              INT00060
       2ZAX(40,40),XBX(40),YBX(40),ZBX(40,40)                           INT00070
119     FORMAT(2I5)                                                     INT00080
131     FORMAT(F7.1)                                                    INT00090
130     FORMAT(F5.1)                                                    INT00100
120     FORMAT(F5.3)                                                    INT00110
132     FORMAT(F10.3)                                                   INT00120
135     FORMAT(F15.5)                                                   INT00130
136     FORMAT(F17.5)                                                   INT00140
C  READ PROBE CALIBRATION DATA                                          INT00150
        READ(2,119) NX,NY                                               INT00160
        DO 121 I=1,NX                                                   INT00170
121     READ(2,130) XA(I)                                               INT00180
        DO 122 I=1,NY                                                   INT00190
122     READ(2,130) YA(I)                                               INT00200
        DO 123 I=1,NX                                                   INT00210
        DO 123 J=1,NY                                                   INT00220
123     READ(2,120) ZA(I,J)                                             INT00230
        DO 221 I=1,NX                                                   INT00240
221     READ(2,130) XB(I)                                               INT00250
        DO 222 I=1,NY                                                   INT00260
222     READ(2,130) YB(I)                                               INT00270
        DO 223 I=1,NX                                                   INT00280
        DO 223 J=1,NY                                                   INT00290
223     READ(2,120) ZB(I,J)                                             INT00300
C  TRANSFORM CALIBRATION MATRICES                                       INT00310
        DO 224 I=1,NY                                                   INT00320
224     XAX(I)=XA(I)                                                    INT00330
        DO 225 I=1,NX                                                   INT00340
225     YAX(I)=XA(I)                                                    INT00350
        DO 226 I=1,NY                                                   INT00360
        DO 226 J=1,NX                                                   INT00370
226     ZAX(I,J)=ZA(J,I)                                                INT00380
        DO 227 I=1,NY                                                   INT00390
227     XBX(I)=YB(I)                                                    INT00400
        DO 228 I=1,NX                                                   INT00410
228     YBX(I)=XB(I)                                                    INT00420
        DO 229 I=1,NY                                                   INT00430
        DO 229 J=1,NX                                                   INT00440
229     ZBX(I,J)=ZB(J,I)                                                INT00450
C  PROBE SETTINGS                                                       INT00460
        YPA=C.C                                                         INT00470
        YPB=55.                                                         INT00480
        XPA=0.0                                                         INT00490
        XPC=20.                                                         INT00500
        XPB=-20.                                                        INT00510
C  PROGRAM CONSTANTS                                                    INT00520
        YRLO=-30.                                                       INT00530
        YRLP=30.                                                        INT00540
        YRIN=-30.                                                       INT00550
        XRLO=-30.                                                       INT00560
        XRLP=30.                                                        INT00570
        XRIN=-30.                                                       INT00580
        ICCPS=1                                                         INT00590
        NOCOPS=1                                                        INT00600
        NOSIM=1                                                         INT00610
        IIT=1                                                           INT00620
        ISCAN=1                                                         INT00630
        RELXPS=0.5                                                      INT00640
C  EXPERIMENT SIMULATION                                               INT00650
401     FORMAT(F15.4)                                                   INT00660
193     READ(5,401) PT                                                  INT00670
        WRITE(6,401) PT                                                 INT00680
        IF(NOSIM.EQ.2) GO TO 500                                        INT00690
        READ(5,401) PS                                                  INT00700
```

```
      WRITE(6,401) PS                                                   INT00710
      READ(5,401) ALFA                                                  INT00720
      WRITE(6,401) ALFA                                                 INT00730
      READ(5,401) PHII                                                  INT00740
      WRITE(6,401) PHII                                                 INT00750
      CALL INTPLT(XA,YA,ZA,ALFA,PHII,NX,NY,CPA)                         INT00760
      ALFC=ALFA-XPC                                                     INT00770
      CALL INTPLT(XA,YA,ZA,ALFC,PHII,NX,NY,CPC)                         INT00780
      ALFD=ALFA-XPD                                                     INT00790
      CALL INTPLT(XA,YA,ZA,ALFD,PHII,NX,NY,CPD)                         INT00800
      PHIB=PHII-YPB                                                     INT00810
      CALL INTPLT(XB,YB,ZB,ALFA,PHIB,NX,NY,CPB)                         INT00820
      PDYN=PT-PS                                                        INT00830
      PPA=CPA*PDYN+PS                                                   INT00840
      PPB=CPB*PDYN+PS                                                   INT00850
      PPC=CPC*PDYN+PS                                                   INT00860
      PPD=CPD*PDYN+PS                                                   INT00870
C READ MEASUREMENTS DATA                                                INT00880
      IF(ISCAN.EQ.1) GO TO 501                                          INT00890
500   READ(5,135) PPA                                                   INT00900
      WRITE(6,136) PPA                                                  INT00910
      READ(5,135) PPB                                                   INT00920
      WRITE(6,136) PPB                                                  INT00930
      READ(5,135) PPC                                                   INT00940
      WRITE(6,136) PPC                                                  INT00950
      READ(5,135) PPD                                                   INT00960
      WRITE(6,136) PPD                                                  INT00970
501   READ(5,135) PSIN                                                  INT00980
      WRITE(6,136) PSIN                                                 INT00990
      IF(ISCAN.EQ.1) GO TO 181                                          INT01000
180   IF(ISCAN.GT.1000) GO TO 305                                       INT01010
      IT=1                                                              INT01020
      ICOPS=1                                                           INT01030
      ISCAN=ISCAN+1                                                     INT01040
      PSIN=PSIN+5.0                                                     INT01050

181   PS=PSIN                                                           INT01070
      PSC=PSIN                                                          INT01080
      PST=PS                                                            INT01090
C CALCULATES PRESSURE COEFFICIENTS                                      INT01100
300   PDYN=PT-PS                                                        INT01110
      CPA=(PPA-PS)/PDYN                                                 INT01120
      CPB=(PPB-PS)/PDYN                                                 INT01130
      CPC=(PPC-PS)/PDYN                                                 INT01140
      CPD=(PPD-PS)/PDYN                                                 INT01150
C CORRECTS PS ASSUMPTION TO ENSURE CP ARE IN CALIBRATION RANGE          INT01160
      IF(NOCTPS.EQ.2) GO TO 301                                         INT01170
      ZAMAX=-10000.                                                     INT01180
      CALL MAXARY(ZA,ZAMAX,NX,NY)                                       INT01190
      IF(CPA.GT.ZAMAX) PSC=1.01*PS                                      INT01200
      IF(CPC.GT.ZAMAX) PSC=1.01*PS                                      INT01210
      IF(CPD.GT.ZAMAX) PSC=1.01*PS                                      INT01220
      ZBMAX=-10000.                                                     INT01230
      CALL MAXARY(ZB,ZBMAX,NX,NY)                                       INT01240
      IF(CPB.GT.ZBMAX) PSC=1.01*PS                                      INT01250
      ZAMIN=10000.                                                      INT01260
      CALL MINARY(ZA,ZAMIN,NX,NY)                                       INT01270
      IF(CPA.LT.ZAMIN) PSC=0.99*PS                                      INT01280
      IF(CPC.LT.ZAMIN) PSC=0.99*PS                                      INT01290
      IF(CPD.LT.ZAMIN) PSC=0.99*PS                                      INT01300
      ZBMIN=10000.                                                      INT01310
      CALL MINARY(ZB,ZBMIN,NX,NY)                                       INT01320
      IF(CPB.LT.ZBMIN) PSC=0.99*PS                                      INT01330
      IF(PSC.EQ.PST) GO TO 301                                          INT01340
      ICOPS=ICOPS+1                                                     INT01350
      IF(ICOPS.GT.10) GO TO 180                                         INT01360
      PS=PSC                                                            INT01370
      PST=PSC                                                           INT01380
      GO TO 300                                                         INT01390
C CALCULATES TWO ALF & PHI ANGLES FOR 'I' & 'II'                        INT01400
```

```
FILE: VDR       FORTRAN  P1                          N       P               S

301     CALL INTSCS(YPA,YPB,YRLC,YRUP,YRIN,                                     INTC1410
       1CPA,CPB,ALF,PHI,XA,YA,ZA,                                               INTC1420
       2XB,YB,ZB,XAX,YAX,ZAX,XBX,YBX,ZBX,NX,NY,IEPS)                            INTC1430
        IEPS1=IEPS                                                              INTC1440
        ALF1=ALF(1)                                                             INTC1450
        ALF2=ALF(2)                                                             INTC1460
        PHI1=PHI(1)                                                             INTC1470
        PHI2=PHI(2)                                                             INTC1480
C CALCULATES TWO ALF & PHI ANGLES FOR 'I' & 'III'                               INTC1490
        CALL INTSCS(XPA,XPC,XRLC,XRUP,XRIN,                                     INTC1500
       1CPA,CPC,PHI,ALF,XAX,YAX,ZAX,                                            INTC1510
       2XAX,YAA,ZAA,XA,YA,ZA,XA,YA,ZA,NY,NX,IEPS)                               INTC1520
        IEPS2=IEPS                                                              INTC1530
        ALF3=ALF(1)                                                             INTC1540
        ALF4=ALF(2)                                                             INTC1550
        PHI3=PHI(1)                                                             INTC1560
        PHI4=PHI(2)                                                             INTC1570
C SELECTS THE PROPER ALF & PHI ANGLES OUT OF THE FOUR VALUES                    INTC1580
        ALF=((ALF1+ALF2+ALF3+ALF4)-AMAX1(ALF1,ALF2,ALF3,                        INTC1590
       1ALF4)-AMIN1(ALF1,ALF2,ALF3,ALF4))/2                                     INTC1600
        PHI=((PHI1+PHI2+PHI3+PHI4)-AMAX1(PHI1,PHI2,PHI3,PHI4)                   INTC1610
       1-AMIN1(PHI1,PHI2,PHI3,PHI4))/2                                          INTC1620
C CALCULATES NEW PS FROM DATA OF POSITION IV                                    INTC1630
        ALFCN=ALF-XPD                                                           INTC1640
        CALL INTFLT(XA,YA,ZA,ALFCN,PHI,NX,NY,CPD)                               INTC1650
        ALFCN=ALF-XPC                                                           INTC1660
        CALL INTFLT(XA,YA,ZA,ALFCN,PHI,NX,NY,CPC)                               INTC1670
        PSN=(PPC*CPD-PPD*CPC)/(CPD-CPC)                                         INTC1680
C CONVERGENCE TESTS                                                             INTC1690
        EPSPS=ABS((PSN-PS)/PS)                                                  INTC1700
        EPSPSC=.000000000009*(PT**2.0)-.00000098*PT                             INTC1710
        IF(IEPS1.EQ.2) EPSPSC=.000000000009*(PT**2.0)-.00000098*PT              INTC1720
        IF(IEPS2.EQ.2) EPSPSC=.000000000009*(PT**2.0)-.00000098*PT              INTC1730
        IF(EPSPS.GT.EPSPSC) GO TO 310                                           INTC1740
        GO TO 303                                                               INTC1750
310     IIT=IIT+1                                                               INTC1760
        IF(IIT.GT.10) GO TO 180                                                 INTC1770
        PS=PS+RELAPS*(PSN-PS)                                                   INTC1780
        PSC=PS                                                                  INTC1790
        PST=PS                                                                  INTC1800
        GO TO 300                                                               INTC1810
303     WRITE(6,171) ALF                                                        INTC1820
171     FORMAT('0',' VELOCITY YAW IS' ,F15.2)                                   INTC1830
        WRITE(6,172) PHI                                                        INTC1840
172     FORMAT(' VELOCITY PITCH IS' ,F15.2)                                     INTC1850
        WRITE(6,173) PSN                                                        INTC1860
173     FORMAT(' STATIC PRESSURE IS' ,F15.3)                                    INTC1870
        WRITE(6,174) PT                                                         INTC1880
174     FORMAT(' TOTAL PRESSURE IS' ,F15.3)                                     INTC1890
        AM=SQRT(((PT/PSN)**0.285714-1.0)*5.0)                                   INTC1900
        WRITE(6,175) AM                                                         INTC1910
175     FORMAT(' MACH NUMBER IS' ,F15.5)                                        INTC1920
CT      CALL IADDN(A)                                                           INTC1930
CT      VIRTIM=(A(5)-IA5C)/76800                                                INTC1940
CT      CPU=(A(6)-IA6C)/76800                                                   INTC1950
CT      WRITE(6,450) VIRTIM                                                     INTC1960
450     FORMAT(' VIRTUAL TIME IS' ,F15.5)                                       INTC1970
CT      WRITE(6,451) CPU                                                        INTC1980
451     FORMAT(' TOTAL CPU IS' ,F15.5)                                          INTC1990
CT      IA5C=A(5)                                                               INTC2000
CT      IA6C=A(6)                                                               INTC2010
        GO TO 193                                                               INTC2020
305     WRITE(6,307)                                                            INTC2030
307     FORMAT(' SCANNED 1000 TIMES')                                           INTC2040
        GO TO 193                                                               INTC2050
        END                                                                     INTC2060
        SUBROUTINE INTSCS(XPA,XPB,XRLC,XRUP,XRIN,                               INTC2070
       1HT1,HT2,RES1,RES2,X1,Y1,Z1,                                             INTC2080
       2X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4,N1,N2,IEPS)                                  INTC2090
        DIMENSION RES1(10),RES2(10),AMRES(10),X1(40),Y1(40),                    INTC2100
```

```
FILE: VOR       FORTRAN P1                    N       P                S

      1Z1(40,40),X2(40),Y2(40),Z2(40,40),X3(40),Y3(40),                  INTC2110
     2Z3(40,40),X4(40),Y4(40),Z4(40,40)                                  INTC2120
       RES1(1)=3000.                                                     INTC2130
       RES1(2)=5000.                                                     INTC2140
       RES2(1)=9000.                                                     INTC2150
       RES2(2)=5000.                                                     INTC2160
       IEPS=1                                                            INTC2170
       ISL=1                                                             INTC2180
       ISR=1                                                             INTC2190
       CAR=1.                                                            INTC2200
       AR2R=AR1N                                                         INTC2210
       GO TO 152                                                         INTC2220
150    AR2R=AR2R+CAR                                                     INTC2230
       AR2RJ=AR2R                                                        INTC2240
152    ARM1=AR2R-ARP1                                                    INTC2250
       ARM2=AR2R-ARP2                                                    INTC2260
C CHECK FOR LOWER CALIBRATION RANGE                                      INTC2270
       IF(ARM1.LT.ARLO) GO TO 150                                        INTC2280
       IF(ARM2.LT.ARLO) GO TO 150                                        INTC2290
       K=1                                                               INTC2300
C CALCULATES INITIAL PENETRATION POINTS                                  INTC2310
       CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)                        INTC2320
       RAJL=ARRES(1)                                                     INTC2330
       RAJR=ARRES(2)                                                     INTC2340
       CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)                        INTC2350
       RBJL=ARRES(1)                                                     INTC2360
       RBJR=ARRES(2)                                                     INTC2370
C CALCULATES SUCCESIVE PENETRATION POINTS                                INTC2380
       DO 140 I=1,10000                                                  INTC2390
       AR2R=AR2R+CAR                                                     INTC2400
       ARM1=AR2R-ARP1                                                    INTC2410
       ARM2=AR2R-ARP2                                                    INTC2420
C CHECK FOR UPPER CALIBRATION RANGE                                      INTC2430
       IF(ARM1.GT.ARUP) GO TO 195                                        INTC2440
       IF(ARM2.GT.ARUP) GO TO 195                                        INTC2450
       CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)                        INTC2460
       RAIL=ARRES(1)                                                     INTC2470
       RAIR=ARRES(2)                                                     INTC2480
       CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)                        INTC2490
       RBIL=ARRES(1)                                                     INTC2500
       RBIR=ARRES(2)                                                     INTC2510
       IF(RAJL.EQ.1000.) ISL=10                                          INTC2520
       IF(RBJL.EQ.1000.) ISL=10                                          INTC2530
       IF(RBJR.EQ.1000.) ISR=10                                          INTC2540
       IF(RAJR.EQ.1000.) ISR=10                                          INTC2550
       IF(RAIL.EQ.1000.) ISL=10                                          INTC2560
       IF(RBIL.EQ.1000.) ISL=10                                          INTC2570
       IF(RBIR.EQ.1000.) ISR=10                                          INTC2580
       IF(RAIR.EQ.1000.) ISR=10                                          INTC2590
C CHECK FOR DIRECT HIT ON LEFT INTERSECTION,EVALUATE IF REQUIRED          INTC2600
       IF(ISL.NE.10) GO TO 164                                           INTC2610
       ISL=1                                                             INTC2620
       GO TO 183                                                         INTC2630
164    IF(RAIL.NE.RBIL) GO TO 180                                        INTC2640
       RES1(K)=RBIL                                                      INTC2650
       RES2(K)=AR2R                                                      INTC2660
       IF(K.EQ.2) GO TO 155                                              INTC2670
       K=2                                                               INTC2680
       GO TO 183                                                         INTC2690
C CHECK FOR INTERMEDIATE LEFT INTERSECTION,EVALUATE IF REQUIRED          INTC2700
180    IF(ISL.NE.10) GO TO 160                                           INTC2710
       ISL=1                                                             INTC2720
       GO TO 183                                                         INTC2730
160    IF((RAJL-RBJL)/(RAIL-RBIL).GT.0.0) GO TO 183                      INTC2740
       AA=(AR2RJ-AR2R)/(RAJL-RAIL)                                       INTC2750
       BA=AR2R-AA*RAIL                                                   INTC2760
       AB=(AR2RJ-AR2R)/(RBJL-RBIL)                                       INTC2770
       BB=AR2R-AB*RBIL                                                   INTC2780
       RES1(K)=(BB-BA)/(AA-AB)                                           INTC2790
       RES2(K)=AA*RES1(K)+BA                                             INTC2800
```

```
FILE: VDR      FORTRAN 21                    N     P              S
      IF(K.EQ.2) GO TO 155                                          INTC2810
      K=2                                                           INTC2820
C CHECK FOR DIRECT HIT OF RIGHT INTERSECTION,EVALUATE IF REQUIRED   INTC2830
183   IF(ISR.NE.1) GO TO 161                                        INTC2840
      ISR=1                                                         INTC2850
      GO TO 163                                                     INTC2860
161   IF(RAIR.NE.RBIR) GO TO 182                                    INTC2870
      RES1(K)=RBIR                                                  INTC2880
      RES2(K)=AR2R                                                  INTC2890
      IF(K.EQ.2) GO TO 155                                          INTC2900
      K=2                                                           INTC2910
      GO TO 163                                                     INTC2920
C CHECK FOR INTERMEDIATE RIGHT INTERSECTION,EVALUATE IF REQUIRED    INTC2930
182   IF(ISR.NE.1) GO TO 162                                        INTC2940
      ISR=1                                                         INTC2950
      GO TO 163                                                     INTC2960
162   IF((RAJR-RBJR)/(RAIR-RBIR).GT.0.0) GO TO 163                  INTC2970
      AA=(AR2J-AR2R)/(RAJR-RAIR)                                    INTC2980
      BA=AR2R-AA*RAIR                                               INTC2990
      AB=(AR2J-AR2R)/(RBJR-RBIR)                                    INTC3000
      BB=AR2R-AB*RBIR                                               INTC3010
      RES1(K)=(BB-BA)/(AA-AB)                                       INTC3020
      RES2(K)=AA*RES1(K)+BA                                         INTC3030
      IF(K.EQ.2) GO TO 155                                          INTC3040
      K=2                                                           INTC3050
C RE-INITIATE                                                       INTC3060
163   AR2J=AR2R                                                     INTC3070
      RAJL=RAIL                                                     INTC3080
      RAJR=RAIR                                                     INTC3090
      RBJL=RBIL                                                     INTC3100
140   RBJR=RBIR                                                     INTC3110
      WRITE(6,210)                                                  INTC3120
210   FORMAT(' LOOP 140 THROUGH,NO POINTS FOUND')                   INTC3130
      GO TO 155                                                     INTC3140
195   IF(RES1(1).EQ.9000.0) GO TO 191                               INTC3150
      IF(RES1(2).EQ.9000.0) GO TO 191                               INTC3160
      IF(RES2(1).EQ.9000.0) GO TO 191                               INTC3170
      IF(RES2(2).EQ.9000.0) GO TO 191                               INTC3180
      GO TO 155                                                     INTC3190
C EVALUATION OF INTERSECTIONS WHEN CURVES ARE ALMOST TANGENT        INTC3200
191   YGC=0.0                                                       INTC3210
      IEPS=2                                                        INTC3220
      CALL PEMPTS(I2,N1,XB,YB,ZB,HT1,YGO,ARRES)                     INTC3230
      R1=ABS(ARRES(2)-ARRES(1))/2                                   INTC3240
      CALL PEMPTS(I2,N1,XA,YA,ZA,HT2,YGO,ARRES)                     INTC3250
      R2=ABS(ARRES(2)-ARRES(1))/2                                   INTC3260
      RES2(1)=((R1**2)-(R2**2)+(ARP2**2))/(2.0*ARP2)                INTC3270
      RES2(2)=RES2(1)                                               INTC3280
      RES1(1)=SQRT(ABS((R1**2)-(RES2(1)**2)))                       INTC3290
      RES1(2)=-RES1(1)                                              INTC3300
155   RETURN                                                        INTC3310
      END                                                           INTC3320
      SUBROUTINE PEMPTS(IX,NY,X,Y,Z,HT,YG,XR)                       INTC3330
      DIMENSION Y(40),ZX(40),Z(40,40),X(40),XR(10)                  INTC3340
      IR=1                                                          INTC3350
C SEARCH FOR J OF LOWER Y LINE                                      INTC3360
      DO 101 I=1,NY                                                 INTC3370
      IF (Y(I).GE.YG) GO TO 102                                     INTC3380
101   CONTINUE                                                      INTC3390
102   J=I-1                                                         INTC3400
C COARSE SCAN FOR ZERO PASS                                         INTC3410
C NEXT 4 LINES ARE EXECUTED IN FIRST SEARCH ONLY                    INTC3420
      YC=(YG-Y(J))/(Y(J+1)-Y(J))                                    INTC3430
      ZX(1)=(Z(1,J+1)-Z(1,J))*YC+Z(1,J)                             INTC3440
      IF(ZX(1).EQ.HT) GO TO 107                                     INTC3450
      I3=2                                                          INTC3460
112   DO 106 I=I3,NX                                                INTC3470
      ZX(I)=(Z(I,J+1)-Z(I,J))*YC+Z(I,J)                             INTC3480
      ZM=(Z(I,J+1)-Z(I-1,J))*(YG-Y(J))/(Y(J+1)-Y(J))+Z(I-1,J)       INTC3490
      IF(ZX(I).EQ.HT) GO TO 108                                     INTC3500
```

```
FILE: VOP      FORTRAN  P1                    N     P              S

      IF((ZX(I-1)-HT)/(ZX(I)-HT).LT.0.0) GO TO 104                          INTC3510
103   CONTINUE                                                              INTC3520
      GO TO 116                                                             INTC3530
C CALCULATION OF PENETRATIONS IN EITHER TRIANGLE                            INTC3540
104   XM=X(I-1)+(X(I)-X(I-1))*(YG-Y(J))/(Y(J+1)-Y(J))                       INTC3550
      IF(ZM.EQ.HT) GO TO 105                                                INTC3560
      IF((ZX(I)-HT)/(ZM-HT).LT.0.0) GO TO 105                               INTC3570
      DD=ABS(HT-ZM)+ABS(ZX(I-1)-HT)                                         INTC3580
      XS=X(I-1)+(XM-X(I-1))*ABS(ZX(I-1)-HT)/DD                              INTC3590
      GO TO 110                                                             INTC3600
105   XS=XM+(X(I)-XM)*ABS(ZM-HT)/(ABS(ZM-HT)+ABS(ZX(I)-HT))                 INTC3610
      GO TO 110                                                             INTC3620
C LOGICAL PENETRATIONS ACCUMULATION                                         INTC3630
107   XR(1)=X(1)                                                            INTC3640
      I=1                                                                   INTC3650
      GO TO 113                                                             INTC3660
108   XR(IR)=X(I)                                                           INTC3670
      GO TO 111                                                             INTC3680
110   XR(IR)=XS                                                             INTC3690
111   IF(IR.EQ.2) GO TO 114                                                 INTC3700
113   IR=2                                                                  INTC3710
      IB=I+1                                                                INTC3720
      GO TO 112                                                             INTC3730
116   IF(IR.EQ.1) GO TO 117                                                 INTC3740
      XR(2)=1000.0                                                          INTC3750
      GO TO 114                                                             INTC3760
117   XR(1)=1000.0                                                          INTC3770
      XR(2)=1000.0                                                          INTC3780
114   RETURN                                                                INTC3790
      END                                                                   INTC3800
      SUBROUTINE INTPL1(X,Y,Z,XG,YG,NX,NY,ZRES)                             INTC3810
C LINEAR INTERPOLATION ON CALIBRATION SURFACE TO EVALUATE                   INTC3820
C CP VALUE AT CO-ORDINATES XG,YG.THE RESULT RETURNED IS ZRES                INTC3830
      DIMENSION X(40),Y(40),Z(40,40)                                        INTC3840
C SEARCH FOR I OF LEFT X LINE                                               INTC3850
      DO 1 IC=I,NX                                                          INTC3860
      IF(X(IC).GE.XG) GO TO 2                                               INTC3870
1     CONTINUE                                                              INTC3880
2     I=IC-1                                                                INTC3890
C SEARCH FOR J OF LOWER Y LINE                                              INTC3900
      DO 3 JC=1,NY                                                          INTC3910
      IF(Y(JC).GE.YG) GO TO 4                                               INTC3920
3     CONTINUE                                                              INTC3930
4     J=JC-1                                                                INTC3940
      AL=(Y(J+1)-Y(J))/(X(I+1)-X(I))                                        INTC3950
      BL=Y(J)-AL*X(I)                                                       INTC3960
      YCR=(AL*XG)+BL                                                        INTC3970
      IF(YG.EQ.YCR) GO TO 7                                                 INTC3980
      IF(YG.LT.YCR) GO TO 5                                                 INTC3990
C RESULTING Z IS IN LOWRER TRIANGLE                                         INTC4000
      X3=X(I+1)                                                             INTC4010
      Y3=Y(J)                                                               INTC4020
      Z3=Z(I+1,J)                                                           INTC4030
      GO TO 6                                                               INTC4040
C RASULTING Z IS IN UPPER TRIANGLE                                          INTC4050
5     X3=X(I)                                                               INTC4060
      Y3=Y(J+1)                                                             INTC4070
      Z3=Z(I,J+1)                                                           INTC4080
C GENERAL CALCULATION:GOOD FOR BOTH TRIANGLES                               INTC4090
6     X1=X(I)                                                               INTC4100
      Y1=Y(J)                                                               INTC4110
      Z1=Z(I,J)                                                             INTC4120
      X2=X(I+1)                                                             INTC4130
      Y2=Y(J+1)                                                             INTC4140
      Z2=Z(I+1,J+1)                                                         INTC4150
      AP=((Z1-Z3)*(Y1-Y2)-(Z1-Z2)*(Y1-Y3))/((X1-X3)*(Y1-Y2)-(X1-X2)         INTC4160
     1*(Y1-Y3))                                                             INTC4170
      BP=((Z1-Z2)-AP*(X1-X2))/(Y1-Y2)                                       INTC4180
      CP=Z1-AP*X1-BP*Y1                                                     INTC4190
      ZRES=AP*XG+BP*YG+CP                                                   INTC4200
```

```
      GO TO 100
C RESULTING Z IS ON DIVIDING LINE
7     ZRES=(Z(I+1,J+1)-Z(I,J))*(XO-X(I))/(X(I+1)-X(I))+Z(I,J)
100   RETURN
      END
      SUBROUTINE MAXARY(Z,ZMAX,NX,NY)
C EVALUATION OF MAXIMUM CP OF CALIBRATION SURFACE MATRIX
      DIMENSION Z(40,40)
      DO 1 I=1,NX
      DO 1 J=1,NY
1     IF(Z(I,J).GE.ZMAX) ZMAX=Z(I,J)
      RETURN
      END
      SUBROUTINE MINARY(Z,ZMIN,NX,NY)
C EVALUATION OF MINIMUM CP OF CALIBRATION SURFACE MATRIX
      DIMENSION Z(40,40)
      DO 1 I=1,NX
      DO 1 J=1,NY
1     IF(Z(I,J).LE.ZMIN) ZMIN=Z(I,J)
      RETURN
      END
C END OF VOR
```

APPENDIX 2

LISTING OF SWVDR

```
  07/08/75  18.45.19

FILE: SAVOR     FORTRAN  P1                    N      P              S

C  START OF SAVOR                                                          INT00010
       DIMENSION ALF(10),PHI(10),XA(40),YA(40),                            INT00020
      1ZA(40,40),XB(40),YB(40),ZB(40,40),XAX(40),YAX(40),                  INT00030
      2ZAX(40,40),XBX(40),YBX(40),ZBX(40,40)                               INT00040
119    FORMAT(2I5)                                                         INT00050
131    FORMAT(F7.1)                                                        INT00060
130    FORMAT(F5.1)                                                        INT00070
120    FORMAT(F8.5)                                                        INT00080
132    FORMAT(F10.5)                                                       INT00090
135    FORMAT(F15.5)                                                       INT00100
136    FORMAT(F17.5)                                                       INT00110
C  READ PROBE CALIBRATION DATA                                             INT00120
       READ(2,119) NX,NY                                                   INT00130
       DO 121 I=1,NX                                                       INT00140
       READ(2,130) XA(I)                                                   INT00150
121    CONTINUE                                                            INT00160
       DO 122 I=1,NY                                                       INT00170
       READ(2,130) YA(I)                                                   INT00180
122    CONTINUE                                                            INT00190
       DO 123 I=1,NX                                                       INT00200
       DO 123 J=1,NY                                                       INT00210
       READ(2,120) ZA(I,J)                                                 INT00220
123    CONTINUE                                                            INT00230
       DO 221 I=1,NX                                                       INT00240
       READ(2,130) XB(I)                                                   INT00250
221    CONTINUE                                                            INT00260
       DO 222 I=1,NY                                                       INT00270
       READ(2,130) YB(I)                                                   INT00280
222    CONTINUE                                                            INT00290
       DO 223 I=1,NX                                                       INT00300
       DO 223 J=1,NY                                                       INT00310
       READ(2,120) ZB(I,J)                                                 INT00320
223    CONTINUE                                                            INT00330
C  TRANSFORM CALIBRATION MATRICES                                          INT00340
       DO 224 I=1,NY                                                       INT00350
224    XAX(I)=YA(I)                                                        INT00360
       DO 225 I=1,NX                                                       INT00370
225    YAX(I)=XA(I)                                                        INT00380
       DO 226 I=1,NY                                                       INT00390
       DO 226 J=1,NX                                                       INT00400
226    ZAX(I,J)=ZA(J,I)                                                    INT00410
       DO 227 I=1,NY                                                       INT00420
227    XBX(I)=YB(I)                                                        INT00430
       DO 228 I=1,NX                                                       INT00440
228    YBX(I)=XB(I)                                                        INT00450
       DO 229 I=1,NY                                                       INT00460
       DO 229 J=1,NX                                                       INT00470
229    ZBX(I,J)=ZB(J,I)                                                    INT00480
C  PROBE SETTINGS                                                          INT00490
       YPA=0.0                                                             INT00500
       YPB=55.                                                             INT00510
       XPA=0.0                                                             INT00520
       XPC=20.                                                             INT00530
       XPD=-20.                                                            INT00540
C  PROGRAM CONSTANTS                                                       INT00550
       YRLO=-30.                                                           INT00560
       YRUP=30.                                                            INT00570
       YRIN=-30.                                                           INT00580
       XRLO=-30.                                                           INT00590
       XRUP=30.                                                            INT00600
       XRIN=-30.                                                           INT00610
       ICOPS=1                                                             INT00620
       NCCCPS=2                                                            INT00630
       NOSIM=2                                                             INT00640
       IIT=1                                                               INT00650
       ISCAN=1                                                             INT00660
       RELXPS=0.5                                                          INT00670
C  EXPERIMENT SIMULATION                                                   INT00680
193    CONTINUE                                                            INT00690
401    FORMAT(F15.4)                                                       INT00700
```

```
FILE: SWVOR    FORTRAN  P1                    -    N     P          S
      READ(5,401) PT                                                    INTC0710
      WRITE(6,401) PT                                                   INTC0720
      IF(NOSIM.EQ.2) GO TO 500                                          INTC0730
      READ(5,401) PS                                                    INTC0740
      WRITE(6,401) PS                                                   INTC0750
      READ(5,401) ALFA                                                  INTC0760
      WRITE(6,401) ALFA                                                 INTC0770
      READ(5,401) PHII                                                  INTC0780
      WRITE(6,401) PHII                                                 INTC0790
      CALL INTPLT(XA,YA,ZA,ALFA,PHII,NX,NY,CPA)                         INTC0800
      ALFC=ALFA-APC                                                     INTC0810
      CALL INTPLT(XA,YA,ZA,ALFC,PHII,NX,NY,CPC)                         INTC0820
      ALFD=ALFA-APD                                                     INTC0830
      CALL INTPLT(XA,YA,ZA,ALFD,PHII,NX,NY,CPD)                         INTC0840
      PHIB=PHII-YPB                                                     INTC0850
      CALL INTPLT(XB,YB,ZB,ALFA,PHIB,NX,NY,CPB)                         INTC0860
      PDYN=PT-PS                                                        INTC0870
      PPA=CPA*PDYN+PS                                                   INTC0880
      PPB=CPB*PDYN+PS                                                   INTC0890
      PPC=CPC*PDYN+PS                                                   INTC0900
      PPD=CPD*PDYN+PS                                                   INTC0910
C READ MEASUREMENTS DATA                                                INTC0920
      IF(NOSIM.EQ.1) GO TO 501                                          INTC0930
500   READ(5,135) PPA                                                   INTC0940
      WRITE(6,136) PPA                                                  INTC0950
      READ(5,135) PPB                                                   INTC0960
      WRITE(6,136) PPB                                                  INTC0970
      READ(5,135) PPC                                                   INTC0980
      WRITE(6,136) PPC                                                  INTC0990
      READ(5,135) PPD                                                   INTC1000
      WRITE(6,136) PPD                                                  INTC1010
501   READ(5,135) PSIM                                                  INTC1020
      WRITE(6,136) PSIM                                                 INTC1030
      IF(ISCAN.EQ.1) GO TO 181                                          INTC1040
180   IF(ISCAN.GT.1000) GO TO 305                                       INTC1050
      III=1                                                             INTC1060
      ICCPS=1                                                           INTC1070
      ISCAN=ISCAN+1                                                     INTC1080
      PSIN=PSIN+5.0                                                     INTC1090
      WRITE(6,136) PSIN                                                 INTC1100
181   CONTINUE                                                          INTC1110
      PS=PSIN                                                           INTC1120
      PSC=PSIN                                                          INTC1130
      PST=PS                                                            INTC1140
C CALCULATES PRESSURE COEFFICIENTS                                      INTC1150
300   CONTINUE                                                          INTC1160
      PDYN=PT-PS                                                        INTC1170
      CPA=(PPA-PS)/PDYN                                                 INTC1180
      CPB=(PPB-PS)/PDYN                                                 INTC1190
      CPC=(PPC-PS)/PDYN                                                 INTC1200
      CPD=(PPD-PS)/PDYN                                                 INTC1210
C CORRECTS PS ASSUMPTION TO ENSURE CP ARE IN CALIBRATION RANGE          INTC1220
      IF(NOCOPS.EQ.2) GO TO 301                                         INTC1230
      ZAMAX=-10000.                                                     INTC1240
      CALL MAXARY(ZA,ZAMAX,NX,NY)                                       INTC1250
      IF(CPA.GT.ZAMAX) PSC=1.01*PS                                      INTC1260
      IF(CPC.GT.ZAMAX) PSC=1.01*PS                                      INTC1270
      IF(CPD.GT.ZAMAX) PSC=1.01*PS                                      INTC1280
      ZBMAX=-10000.                                                     INTC1290
      CALL MAXARY(ZB,ZBMAX,NX,NY)                                       INTC1300
      IF(CPB.GT.ZBMAX) PSC=1.01*PS                                      INTC1310
      ZAMIN=10000.                                                      INTC1320
      CALL MINARY(ZA,ZAMIN,NX,NY)                                       INTC1330
      IF(CPA.LT.ZAMIN) PSC=0.99*PS                                      INTC1340
      IF(CPC.LT.ZAMIN) PSC=0.99*PS                                      INTC1350
      IF(CPD.LT.ZAMIN) PSC=0.99*PS                                      INTC1360
      ZBMIN=10000.                                                      INTC1370
      CALL MINARY(ZB,ZBMIN,NX,NY)                                       INTC1380
      IF(CPB.LT.ZBMIN) PSC=0.99*PS                                      INTC1390
      IF(PSC.EQ.PST) GO TO 301                                          INTC1400
```

```
      ICOPS=ICOPS+1                                                     INTC1410
      IF(ICOPS.GT.10) GO TO 180                                         INTC1420
      PS=PSC                                                            INTC1430
      PST=PSC                                                           INTC1440
      GO TO 300                                                         INTC1450
301   CONTINUE                                                          INTC1460
C CALCULATES TWO ALF & PHI ANGLES FOR '1' & 'II'                        INTC1470
      CALL INTSCS(XPA,YPA,XRLO,XRUP,YRIN,                               INTC1480
     1CPA,CPB,ALF,PHI,XA,YA,ZA,                                         INTC1490
     2XB,YB,ZB,XAX,YAX,ZAX,XBX,YBX,ZBX,NX,NY)                           INTC1500
      ALF1=ALF(1)                                                       INTC1510
      ALF2=ALF(2)                                                       INTC1520
      PHI1=PHI(1)                                                       INTC1530
      PHI2=PHI(2)                                                       INTC1540
C CALCULATES TWO ALF & PHI ANGLES FOR '1' & 'III'                       INTC1550
      CALL INTSCS(XPA,XPC,XRLO,XRUP,XRIN,                               INTC1560
     1CPA,CPC,PHI,ALF,XAX,YAX,ZAX,                                      INTC1570
     2XAX,YAX,ZAX,XA,YA,ZA,XA,YA,ZA,NY,NX)                              INTC1580
      ALF3=ALF(1)                                                       INTC1590
      ALF4=ALF(2)                                                       INTC1600
      PHI3=PHI(1)                                                       INTC1610
      PHI4=PHI(2)                                                       INTC1620
C SELECTS THE PROPER ALF & PHI ANGLES OUT OF THE FOUR VALUES            INTC1630
      FALF=((ALF1+ALF2+ALF3+ALF4)-AMAX1(ALF1,ALF2,ALF3,                 INTC1640
     1ALF4)-AMIN1(ALF1,ALF2,ALF3,ALF4))/2                               INTC1650
      FPHI=((PHI1+PHI2+PHI3+PHI4)-AMAX1(PHI1,PHI2,PHI3,PHI4)            INTC1660
     1-AMIN1(PHI1,PHI2,PHI3,PHI4))/2                                    INTC1670
C CALCULATES NEW PS FROM DATA OF POSITION D AND CORRECTS PT             INTC1680
      ALFDN=FALF-XPD                                                    INTC1690
      CALL INTPLT(XA,YA,ZA,ALFDN,FPHI,NX,NY,CPD)                        INTC1700
      ALFCN=FALF-XPC                                                    INTC1710
      CALL INTPLT(XA,YA,ZA,ALFCN,FPHI,NX,NY,CPC)                        INTC1720
      PSN=(PPC*CPD-PPD*CPC)/(CPD-CPC)                                   INTC1730
907   WRITE(6,911)III,FALF,FPHI,PSN                                     INTC1740
911   FORMAT(' ITERATION VALUES',I5,10X,2F6.2,F10.2)                    INTC1750
C CONVERGENCE TESTS                                                     INTC1760
      EPSPS=ABS((PSN-PS)/PS)                                            INTC1770
      EPSPSG=.000000000009*(PT**2.0)-.00000098*PT                       INTC1780
      IF(IEPS1.EQ.2) EPSPSG=.000000000009*(PT**2.0)-.0000098*PT         INTC1790
      IF(IEPS2.EQ.2) EPSPSG=.000000000009*(PT**2.0)-.0000088*PT         INTC1800
      IF(EPSPS.GT.EPSPSG) GO TO 310                                     INTC1810
      GO TO 303                                                         INTC1820
310   CONTINUE                                                          INTC1830
      III=III+1                                                         INTC1840
      IF(III.GT.10) GO TO 180                                           INTC1850
      PS=PS+RELXPS*(PSN-PS)                                             INTC1860
      PSC=PS                                                            INTC1870
      PST=PS                                                            INTC1880
      GO TO 300                                                         INTC1890
303   WRITE(6,171) FALF                                                 INTC1900
171   FORMAT('0',' VELOCITY YAW IS' ,F15.2)                             INTC1910
      WRITE(6,172) FPHI                                                 INTC1920
172   FORMAT(' VELOCITY PITCH IS' ,F15.2)                               INTC1930
      WRITE(6,173) PSN                                                  INTC1940
173   FORMAT(' STATIC PRESSURE IS' ,F15.3)                              INTC1950
      WRITE(6,174) PT                                                   INTC1960
174   FORMAT(' TOTAL PRESSURE IS' ,F15.3)                               INTC1970
      AM=SQRT(((PT/PSN)**0.285714-1.0)*5.0)                             INTC1980
      WRITE(6,175) AM                                                   INTC1990
175   FORMAT(' MACH NUMBER IS' ,F15.5)                                  INTC2000
      GO TO 195                                                         INTC2010
305   WRITE(6,307)                                                      INTC2020
307   FORMAT(' SCANNED 1000 TIMES')                                     INTC2030
      GO TO 195                                                         INTC2040
      END                                                               INTC2050
      SUBROUTINE INTSCS(ARP1,ARP2,ARLO,ARUP,ARIN,                       INTC2060
     1HT1,HT2,RES1,RES2,X1,Y1,Z1,                                       INTC2070
     2X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4,N1,N2)                                 INTC2080
      DIMENSION RES1(10),RES2(10),ARRES(10),X1(40),Y1(40),              INTC2090
     1Z1(40,40),X2(40),Y2(40),Z2(40,40),X3(40),Y3(40),                  INTC2100
```

```
FILE: SAVER     FORTRAN  P1                    N     P              S

      2Z3(40,40),X4(40),Y4(40),Z4(40,40)                                INTC2110
       RES1(1)=9000.                                                    INTC2120
       RES1(2)=9000.                                                    INTC2130
       RES2(1)=9000.                                                    INTC2140
       RES2(2)=9000.                                                    INTC2150
       ISL=1                                                            INTC2160
       ISR=1                                                            INTC2170
       DAR=1.                                                           INTC2180
       AR2R=ARIN                                                        INTC2190
       GO TO 152                                                        INTC2200
150    AR2R=AR2R+DAR                                                    INTC2210
       AR2J=AR2R                                                        INTC2220
152    ARM1=AR2R-ARP1                                                   INTC2230
       ARM2=AR2R-ARP2                                                   INTC2240
C CHECK FOR LOWER CALIBRATION RANGE                                     INTC2250
       IF(ARM1.LT.ARLO) GO TO 150                                       INTC2260
       IF(ARM2.LT.ARLO) GO TO 150                                       INTC2270
       K=1                                                              INTC2280
C CALCULATES INITIAL PENETRATION POINTS                                 INTC2290
       CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)                       INTC2300
       RAJL=ARRES(1)                                                    INTC2310
       RAJR=ARRES(2)                                                    INTC2320
       CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)                       INTC2330
       RBJL=ARRES(1)                                                    INTC2340
       RBJR=ARRES(2)                                                    INTC2350
C CALCULATES SUCCESSIVE PENETRATION POINTS                              INTC2360
       DO 140 I=1,10000                                                 INTC2370
       AR2R=AR2R+DAR                                                    INTC2380
       ARM1=AR2R-ARP1                                                   INTC2390
       ARM2=AR2R-ARP2                                                   INTC2400
C CHECK FOR UPPER CALIBRATION RANGE                                     INTC2410
       IF(ARM1.GT.ARUP) GO TO 195                                       INTC2420
       IF(ARM2.GT.ARUP) GO TO 195                                       INTC2430
       CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)                       INTC2440
       RAIL=ARRES(1)                                                    INTC2450
       RAIR=ARRES(2)                                                    INTC2460
       CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)                       INTC2470
       RBIL=ARRES(1)                                                    INTC2480
       RBIR=ARRES(2)                                                    INTC2490
       IF(RAJL.EQ.1000.) ISL=10                                         INTC2500
       IF(RBJL.EQ.1000.) ISL=10                                         INTC2510
       IF(RAJR.EQ.1000.) ISR=10                                         INTC2520
       IF(RBJR.EQ.1000.) ISR=10                                         INTC2530
       IF(RAIL.EQ.1000.) ISL=10                                         INTC2540
       IF(RBIL.EQ.1000.) ISL=10                                         INTC2550
       IF(RBIR.EQ.1000.) ISR=10                                         INTC2560
       IF(RAIR.EQ.1000.) ISR=10                                         INTC2570
C CHECK FOR DIRECT HIT ON LEFT INTERSECTION,EVALUATE IF REQUIRED         INTC2580
       IF(ISL.NE.10) GO TO 164                                          INTC2590
       ISL=1                                                            INTC2600
       GO TO 183                                                        INTC2610
164    IF(RAIL.NE.RAIL) GO TO 180                                       INTC2620
       RES1(K)=RBIL                                                     INTC2630
       RES2(K)=AR2R                                                     INTC2640
       IF(K.EQ.2) GO TO 155                                             INTC2650
       K=2                                                              INTC2660
       GO TO 183                                                        INTC2670
C CHECK FOR INTERMEDIATE LEFT INTERSECTION,EVALUATE IF REQUIRED          INTC2680
180    IF(ISL.NE.10) GO TO 160                                          INTC2690
       ISL=1                                                            INTC2700
       GO TO 183                                                        INTC2710
160    IF((RAJL-RBJL)/(RAIL-RBIL).GT.0.0) GO TO 183                     INTC2720
       AA=(AR2RJ-AR2R)/(RAJL-RAIL)                                      INTC2730
       BA=AR2R-AA*RAIL                                                  INTC2740
       AB=(AR2RJ-AR2R)/(RBJL-RBIL)                                      INTC2750
       BB=AR2R-AB*RBIL                                                  INTC2760
       RES1(K)=(BB-BA)/(AA-AB)                                          INTC2770
       RES2(K)=AA*RES1(K)+BA                                            INTC2780
       IF(K.EQ.2) GO TO 155                                             INTC2790
       K=2                                                              INTC2800
```

```
FILE: SWVOR     FORTRAN  P1                    N     P              S

C CHECK FOR DIRECT HIT ON RIGHT INTERSECTION,EVALUATE IF REQUIRED        INTC2810
183     IF(ISR.NE.10) GO TO 161                                          INTC2820
        ISR=1                                                            INTC2830
        GO TO 163                                                        INTC2840
161     IF(RAIR.NE.RBIR) GO TO 162                                       INTC2850
        RES1(K)=RBIR                                                     INTC2860
        RES2(K)=AR2R                                                     INTC2870
        IF(K.EQ.2) GO TO 155                                             INTC2880
        K=2                                                              INTC2890
        GO TO 163                                                        INTC2900
C CHECK FOR INTERMEDIATE RIGHT INTERSECTION,EVALUATE IF REQUIRED         INTC2910
162     IF(ISR.NE.10) GO TO 162                                          INTC2920
        ISR=1                                                            INTC2930
        GO TO 163                                                        INTC2940
162     IF((RAJR-RBJR)/(RAIR-RBIR).GT.0.0) GO TO 163                     INTC2950
        AA=(AR2RJ-AR2R)/(RAJR-RAIR)                                      INTC2960
        BA=AR2R-AA*RAIR                                                  INTC2970
        AB=(AR2RJ-AR2R)/(RBJR-RBIR)                                      INTC2980
        BB=AR2R-AB*RBIR                                                  INTC2990
        RES1(K)=(BB-BA)/(AA-AB)                                          INTC3000
        RES2(K)=AA*RES1(K)+BA                                            INTC3010
        IF(K.EQ.2) GO TO 155                                             INTC3020
        K=2                                                              INTC3030
C RE-INITIATE                                                            INTC3040
163     AR2RJ=AR2R                                                       INTC3050
        RAJL=RAIL                                                        INTC3060
        RAJR=RAIR                                                        INTC3070
        RBJL=RBIL                                                        INTC3080
        RBJR=RBIR                                                        INTC3090
140     CONTINUE                                                         INTC3100
        WRITE(6,210)                                                     INTC3110
210     FORMAT(' LOOP 140 THROUGH,NO POINTS FOUND')                      INTC3120
        GO TO 155                                                        INTC3130
195     IF(RES1(1).EQ.9000.0) GO TO 191                                  INTC3140
        IF(RES1(2).EQ.9000.0) GO TO 191                                  INTC3150
        IF(RES2(1).EQ.9000.0) GO TO 191                                  INTC3160
        IF(RES2(2).EQ.9000.0) GO TO 191                                  INTC3170
        GO TO 155                                                        INTC3180
C EVALUATION OF INTERSECTIONS WHEN CURVES ARE ALMOST TANGENT             INTC3190
191     YGO=0.0                                                          INTC3200
        CALL PENPTS(N2,N1,X3,Y3,Z3,HT1,YGO,ARRES)                        INTC3210
        R1=ABS(ARRES(2)-ARRES(1))/2                                      INTC3220
        CALL PENPTS(N2,N1,X4,Y4,Z4,HT2,YGO,ARRES)                        INTC3230
        R2=ABS(ARRES(2)-ARRES(1))/2                                      INTC3240
        RES2(1)=((R1**2)-(R2**2)+(ARP2**2))/(2.0*ARP2)                   INTC3250
        RES2(2)=RES2(1)                                                  INTC3260
        RES1(1)=SQRT(ABS((R1**2)-(RES2(1)**2)))                          INTC3270
        RES1(2)=-RES1(1)                                                 INTC3280
155     RETURN                                                           INTC3290
        END                                                              INTC3300
        SUBROUTINE PENPTS(NX,NY,X,Y,Z,HT,YG,XR)                          INTC3310
        DIMENSION Y(40),ZX(40),Z(40,40),X(40),XR(10)                     INTC3320
        IR=1                                                             INTC3330
C SEARCH FOR J OF LOWER Y LINE                                           INTC3340
        DO 101 I=1,NY                                                    INTC3350
        IF (Y(I).GE.YG) GO TO 102                                        INTC3360
101     CONTINUE                                                         INTC3370
102     J=I-1                                                            INTC3380
C COARSE SCAN FOR ZERO PASS                                              INTC3390
C NEXT 4 LINES ARE EXECUTED IN FIRST SEARCH ONLY                         INTC3400
        YC=(YG-Y(J))/(Y(J+1)-Y(J))                                       INTC3410
        ZX(1)=(Z(1,J+1)-Z(1,J))*YC+Z(1,J)                                INTC3420
        IF(ZX(1).EQ.HT) GO TO 107                                        INTC3430
        IB=2                                                             INTC3440
112     DO 103 I=IB,NX                                                   INTC3450
        ZX(I)=(Z(I,J+1)-Z(I,J))*YC+Z(I,J)                                INTC3460
        ZM=(Z(I,J+1)-Z(I-1,J))*(YG-Y(J))/(Y(J+1)-Y(J))+Z(I-1,J)          INTC3470
        IF(ZX(I).EQ.HT) GO TO 108                                        INTC3480
        IF((ZX(I-1)-HT)/(ZX(I)-HT).LT.0.0) GO TO 104                     INTC3490
103     CONTINUE                                                         INTC3500
```

```
FILE: SWVOR    FORTRAN  P1                    N      P              S

      GO TO 116                                                          INTC3510
C CALCULATION OF PENETRATIONS IN EITHER TRIANGLE                         INTC3520
104   XM=X(I-1)+(X(I)-X(I-1))*(YG-Y(J))/(Y(J+1)-Y(J))                   INTC3530
      IF(ZM.EQ.HT) GO TO 105                                             INTC3540
      IF((ZX(I)-HT)/(ZM-HT).LT.0.0) GO TO 105                            INTC3550
      DD=ABS(HT-ZM)+ABS(ZX(I-1)-HT)                                      INTC3560
      XS=X(I-1)+(XM-X(I-1))*ABS(ZX(I-1)-HT)/DD                           INTC3570
      GO TO 110                                                          INTC3580
105   XS=XM+(X(I)-XM)*ABS(ZM-HT)/(ABS(ZM-HT)+ABS(ZX(I)-HT))              INTC3590
      GO TO 110                                                          INTC3600
C LOGICAL PENETRATIONS ACCUMULATION                                      INTC3610
107   XR(1)=X(1)                                                         INTC3620
      I=1                                                                INTC3630
      GO TO 113                                                          INTC3640
108   XR(IR)=X(I)                                                        INTC3650
      GO TO 111                                                          INTC3660
110   XR(IR)=XS                                                          INTC3670
111   IF(IR.EQ.2) GO TO 114                                              INTC3680
113   IR=2                                                               INTC3690
      IB=I+1                                                             INTC3700
      GO TO 112                                                          INTC3710
116   IF(IF.EQ.1) GO TO 117                                              INTC3720
      XR(2)=1000.0                                                       INTC3730
      GO TO 114                                                          INTC3740
117   XR(1)=1000.0                                                       INTC3750
      XR(2)=1000.0                                                       INTC3760
114   RETURN                                                             INTC3770
      END                                                                INTC3780
      SUBROUTINE INTPLT(X,Y,Z,XG,YG,NX,NY,ZRES)                          INTC3790
C LINEAR INTERPOLATION ON CALIBRATION SURFACE TO EVALUATE                INTC3800
C CP VALUE AT CO-ORDINATES XG,YG. THE RESULT RETURNED IS ZRES            INTC3810
      DIMENSION X(40),Y(40),Z(40,40)                                     INTC3820
C SEARCH FOR I OF LEFT X LINE                                            INTC3830
      DO 1 IC=1,NX                                                       INTC3840
      IF(X(IC).GE.XG) GO TO 2                                            INTC3850
1     CONTINUE                                                           INTC3860
2     I=IC-1                                                             INTC3870
C SEARCH FOR J OF LOWER Y LINE                                           INTC3880
      DO 3 JC=1,NY                                                       INTC3890
      IF(Y(JC).GE.YG) GO TO 4                                            INTC3900
3     CONTINUE                                                           INTC3910
4     J=JC-1                                                             INTC3920
      AL=(Y(J+1)-Y(J))/(X(I+1)-X(I))                                     INTC3930
      BL=Y(J)-AL*X(I)                                                    INTC3940
      YCR=(AL*XG)+BL                                                     INTC3950
      IF(YG.EQ.YCR) GO TO 7                                              INTC3960
      IF(YG.GT.YCR) GO TO 5                                              INTC3970
C RESULTING Z IS IN LOWRER TRIANGLE                                      INTC3980
      X3=X(I+1)                                                          INTC3990
      Y3=Y(J)                                                            INTC4000
      Z3=Z(I+1,J)                                                        INTC4010
      GO TO 6                                                            INTC4020
C RESULTING Z IS IN UPPER TRIANGLE                                       INTC4030
5     X3=X(I)                                                            INTC4040
      Y3=Y(J+1)                                                          INTC4050
      Z3=Z(I,J+1)                                                        INTC4060
C GENERAL CALCULATION:GOOD FOR BOTH TRIANGLES                            INTC4070
6     X1=X(I)                                                            INTC4080
      Y1=Y(J)                                                            INTC4090
      Z1=Z(I,J)                                                          INTC4100
      X2=X(I+1)                                                          INTC4110
      Y2=Y(J+1)                                                          INTC4120
      Z2=Z(I+1,J+1)                                                      INTC4130
      AP=((Z1-Z3)*(Y1-Y2)-(Z1-Z2)*(Y1-Y3))/((X1-X3)*(Y1-Y2)-(X1-X2)      INTC4140
     1*(Y1-Y3))                                                          INTC4150
      BP=((Z1-Z2)-AP*(X1-X2))/(Y1-Y2)                                    INTC4160
      CP=Z1-AP*X1-BP*Y1                                                  INTC4170
      ZRES=AP*XG+BP*YG+CP                                                INTC4180
      GO TO 100                                                          INTC4190
C RESULTING Z IS ON DIVIDING LINE                                        INTC4200
```

```
FILE: SHVOR     FORTRAN  P1  .                    N     P              S

7       ZRES=(Z(I+1,J+1)-Z(I,J))*(XG-X(I))/(X(I+1)-X(I))+Z(I,J)                  INTC4210
100     RETURN                                                                   INTC4220
        END                                                                      INTC4230
        SUBROUTINE MAXARY(Z,ZMAX,NX,NY)                                          INTC4240
C EVALUATION OF MAXIMUM OF OF CALIBRATION SURFACE MATRIX                         INTC4250
        DIMENSION Z(40,40)                                                       INTC4260
        DO 1 I=1,NX                                                              INTC4270
        DO 1 J=1,NY                                                              INTC4280
1       IF(Z(I,J).GE.ZMAX) ZMAX=Z(I,J)                                           INTC4290
        RETURN                                                                   INTC4300
        END                                                                      INTC4310
        SUBROUTINE MINARY(Z,ZMIN,NX,NY)                                          INTC4320
C EVALUATION OF MINIMUM OF OF CALIBRATION SURFACE MATRIX                         INTC4330
        DIMENSION Z(40,40)                                                       INTC4340
        DO 1 I=1,NX                                                              INTC4350
        DO 1 J=1,NY                                                              INTC4360
1       IF(Z(I,J).LE.ZMIN) ZMIN=Z(I,J)                                           INTC4370
        RETURN                                                                   INTC4380
        END                                                                      INTC4390
C END OF SHVOR                                                                   INTC4400
                                                                                 INTC4410
```

APPENDIX 3

LISTING OF WVDR

```
  07/08/79  18.45.58

FILE: WVER     FORTRAN  P1                    N      P              S

C  START OF WVER                                                          INTC0010
       DIMENSION ALF(10),PHI(10),XA(40),YA(40),                           INTC0020
      1ZA(40,40),XB(40),YB(40),ZB(40,40),XAX(40),YAX(40),                 INTC0030
      2ZAX(40,40),XBX(40),YBX(40),ZBX(40,40)                              INTC0040
119    FORMAT(2I)                                                         INTC0050
131    FORMAT(F7.1)                                                       INTC0060
130    FORMAT(F5.1)                                                       INTC0070
120    FORMAT(F6.5)                                                       INTC0080
132    FORMAT(F10.5)                                                      INTC0090
135    FORMAT(F15.5)                                                      INTC0100
136    FORMAT(F17.5)                                                      INTC0110
C  READ PROBE CALIBRATION DATA                                            INTC0120
       READ(2,119) NX,NY                                                  INTC0130
       DO 121 I=1,NX                                                      INTC0140
       READ(2,130) XA(I)                                                  INTC0150
121    CONTINUE                                                           INTC0160
       DO 122 I=1,NY                                                      INTC0170
       READ(2,130) YA(I)                                                  INTC0180
122    CONTINUE                                                           INTC0190
       DO 123 I=1,NX                                                      INTC0200
       DO 123 J=1,NY                                                      INTC0210
       READ(2,120) ZA(I,J)                                                INTC0220
123    CONTINUE                                                           INTC0230
       DO 221 I=1,NX                                                      INTC0240
       READ(2,130) XB(I)                                                  INTC0250
221    CONTINUE                                                           INTC0260
       DO 222 I=1,NY                                                      INTC0270
       READ(2,130) YB(I)                                                  INTC0280
222    CONTINUE                                                           INTC0290
       DO 223 I=1,NX                                                      INTC0300
       DO 223 J=1,NY                                                      INTC0310
       READ(2,120) ZB(I,J)                                                INTC0320
223    CONTINUE                                                           INTC0330
C  TRANSFORM CALIBRATION MATRICES                                         INTC0340
       DO 224 I=1,NY                                                      INTC0350
224    XAX(I)=YA(I)                                                       INTC0360
       DO 225 I=1,NX                                                      INTC0370
225    YAX(I)=XA(I)                                                       INTC0380
       DO 226 I=1,NY                                                      INTC0390
       DO 226 J=1,NX                                                      INTC0400
226    ZAX(I,J)=ZA(J,I)                                                   INTC0410
       DO 227 I=1,NY                                                      INTC0420
227    XBX(I)=YB(I)                                                       INTC0430
       DO 228 I=1,NX                                                      INTC0440
228    YBX(I)=XB(I)                                                       INTC0450
       DO 229 I=1,NY                                                      INTC0460
       DO 229 J=1,NX                                                      INTC0470
229    ZBX(I,J)=ZB(J,I)                                                   INTC0480
C  PROBE SETTINGS                                                         INTC0490
       YPA=0.0                                                            INTC0500
       YPB=55.                                                            INTC0510
       XPA=0.0                                                            INTC0520
       XPC=20.                                                            INTC0530
       XPD=-20.                                                           INTC0540
C  PROGRAM CONSTANTS                                                      INTC0550
       YRLC=-30.                                                          INTC0560
       YRUP=30.                                                           INTC0570
       YRIN=-30.                                                          INTC0580
       XRLC=-30.                                                          INTC0590
       XRUP=30.                                                           INTC0600
       XRIN=-30.                                                          INTC0610
       ICOPS=1                                                            INTC0620
      NOSOPS=2                                                            INTC0630
       NOSIM=1                                                            INTC0640
       III=1                                                              INTC0650
       ISCAN=1                                                            INTC0660
       RELXPS=0.0                                                         INTC0670
C  EXPERIMENT SIMULATION                                                  INTC0680
199    CONTINUE                                                           INTC0690
401    FORMAT(F15.4)                                                      INTC0700
```

```
FILE: NVDR     FORTRAN  P1                    N     P              S
      READ(5,401) PT                                                    INT00710
      WRITE(6,401) PT                                                   INT00720
      IF(NCSIM.EQ.2) GO TO 500                                          INT00730
      READ(5,401) PS                                                    INT00740
      WRITE(6,401) PS                                                   INT00750
      READ(5,401) ALFA                                                  INT00760
      WRITE(6,401) ALFA                                                 INT00770
      READ(5,401) PHII                                                  INT00780
      WRITE(6,401) PHII                                                 INT00790
      CALL INTPLT(XA,YA,ZA,ALFA,PHII,NX,NY,CPA)                         INT00800
      ALFC=ALFA-XPC                                                     INT00810
      CALL INTPLT(XA,YA,ZA,ALFC,PHII,NX,NY,CPC)                         INT00820
      ALFD=ALFA-XPD                                                     INT00830
      CALL INTPLT(XA,YA,ZA,ALFD,PHII,NX,NY,CPD)                         INT00840
      PHIB=PHII-YPB                                                     INT00850
      CALL INTPLT(XB,YB,ZB,ALFA,PHIB,NX,NY,CPB)                         INT00860
      WRITE(6,402) CPA,CPB,CPC,CPD                                      INT00870
402   FORMAT(' TRUE CPS',4F10.5)                                        INT00880
      PDYN=PT-PS                                                        INT00890
      PPA=CPA*PDYN+PS                                                   INT00900
      PPB=CPB*PDYN+PS                                                   INT00910
      PPC=CPC*PDYN+PS                                                   INT00920
      PPD=CPD*PDYN+PS                                                   INT00930
      WRITE(6,403) PPA,PPB,PPC,PPD                                      INT00940
403   FORMAT(' SIMUL PRS',4F10.2)                                       INT00950
C READ MEASUREMENTS DATA                                                INT00960
      IF(NCSIM.EQ.1) GO TO 501                                          INT00970
500   READ(5,135) PPA                                                   INT00980
      WRITE(6,136) PPA                                                  INT00990
      READ(5,135) PPB                                                   INT01000
      WRITE(6,136) PPB                                                  INT01010
      READ(5,135) PPC                                                   INT01020
      WRITE(6,136) PPC                                                  INT01030
      READ(5,135) PPD                                                   INT01040
      WRITE(6,136) PPD                                                  INT01050
501   READ(5,135) PSIN                                                  INT01060
      WRITE(6,136) PSIN                                                 INT01070
      IF(ISCAN.EQ.1) GO TO 181                                          INT01080
180   IF(ISCAN.GT.1000) GO TO 305                                       INT01090
      IIT=1                                                             INT01100
      ICOPS=1                                                           INT01110
      ISCAN=ISCAN+1                                                     INT01120
      PSIN=PSIN+5.0                                                     INT01130
      WRITE(6,136) PSIN                                                 INT01140
181   CONTINUE                                                          INT01150
      PS=PSIN                                                           INT01160
      PSC=PS                                                            INT01170
      PST=PS                                                            INT01180
C CALCULATES PRESSURE COEFFICIENTS                                      INT01190
300   CONTINUE                                                          INT01200
      WRITE(6,940) IIT                                                  INT01210
940   FORMAT('0',' ITERATION NO. ',I5)                                  INT01220
      PDYN=PT-PS                                                        INT01230
      CPA=(PPA-PS)/PDYN                                                 INT01240
      CPB=(PPB-PS)/PDYN                                                 INT01250
      CPC=(PPC-PS)/PDYN                                                 INT01260
      CPD=(PPD-PS)/PDYN                                                 INT01270
908   WRITE(6,912) CPA,CPB,CPC,CPD                                      INT01280
912   FORMAT(' CALC CPS',4F10.5)                                        INT01290
C CORRECTS PS ASSUMPTION IF PRESSURE CP ARE IN CALIBRATION RANGE        INT01300
      IF(NCCPS.EQ.2) GO TO 501                                          INT01310
      ZAMAX=-10000.                                                     INT01320
      CALL MAXIMY(ZA,ZAMAX,NX,NY)                                       INT01330
      IF(CPA.GT.ZAMAX) PSC=1.01*PS                                      INT01340
      IF(CPC.GT.ZAMAX) PSC=1.01*PS                                      INT01350
      IF(CPD.GT.ZAMAX) PSC=1.01*PS                                      INT01360
      ZBMAX=-10000.                                                     INT01370
      CALL MAXIMY(ZB,ZBMAX,NX,NY)                                       INT01380
      IF(CPB.GT.ZBMAX) PSC=1.01*PS                                      INT01390
      ZAMIN=10000.                                                      INT01400
```

```
FILE: WVDR     FORTRAN  P1                    N     P              S
      CALL MINARY(ZA,ZAMIN,NX,NY)                                          INT01410
      IF(CPA.LT.ZAMIN) PSC=0.99*PS                                         INT01420
      IF(CPC.LT.ZAMIN) PSC=0.99*PS                                         INT01430
      IF(CPD.LT.ZAMIN) PSC=0.99*PS                                         INT01440
      ZBMIN=10000.                                                         INT01450
      CALL MINARY(ZB,ZBMIN,NX,NY)                                          INT01460
      IF(CPB.LT.ZBMIN) PSC=0.99*PS                                         INT01470
      IF(PSC.EQ.PST) GO TO 301                                             INT01480
      ICCPS=ICCPS+1                                                        INT01490
963   WRITE(6,964) ICCPS                                                   INT01500
964   FORMAT(' ICCPS IS' ,I5)                                              INT01510
      IF(ICCPS.GT.10) GO TO 180                                            INT01520
      PS=PSC                                                               INT01530
      PST=PSC                                                              INT01540
926   WRITE(6,921) PSC                                                     INT01550
921   FORMAT(' PSC IS     ',F20.5)                                         INT01560
      GO TO 300                                                            INT01570
301   CONTINUE                                                             INT01580
C CALCULATES TWO ALF & PHI ANGLES FOR 'I' & 'II'                           INT01590
      CALL INTSCS(XPA,XPB,XRLO,XRUP,XRIN,                                  INT01600
     1CPA,CPB,ALF,PHI,XA,YA,ZA,                                            INT01610
     2XB,YB,ZB,XAX,YAX,ZAX,XBX,YBX,ZBX,NX,NY)                              INT01620
      ALF1=ALF(1)                                                          INT01630
      ALF2=ALF(2)                                                          INT01640
      PHI1=PHI(1)                                                          INT01650
      PHI2=PHI(2)                                                          INT01660
C CALCULATES TWO ALF & PHI ANGLES FOR 'I' & 'III'                          INT01670
      CALL INTSCS(XPA,XPC,XRLO,XRUP,XRIN,                                  INT01680
     1CPA,CPC,PHI,ALF,XAX,YAX,ZAX,                                         INT01690
     2XAX,YAX,ZAX,XA,YA,ZA,XA,YA,ZA,NY,NX)                                 INT01700
      ALF3=ALF(1)                                                          INT01710
      ALF4=ALF(2)                                                          INT01720
      PHI3=PHI(1)                                                          INT01730
      PHI4=PHI(2)                                                          INT01740
C SELECTS THE PROPER ALF & PHI ANGLES OUT OF THE FOUR VALUES               INT01750
      FALF=((ALF1+ALF2+ALF3+ALF4)-AMAX1(ALF1,ALF2,ALF3,                    INT01760
     1ALF4)-AMIN1(ALF1,ALF2,ALF3,ALF4))/2                                  INT01770
      FPHI=((PHI1+PHI2+PHI3+PHI4)-AMAX1(PHI1,PHI2,PHI3,PHI4)               INT01780
     1-AMIN1(PHI1,PHI2,PHI3,PHI4))/2                                       INT01790
900   WRITE(6,901)FALF,FPHI                                                INT01800
901   FORMAT(' SELECTED',2F10.2)                                           INT01810
C CALCULATES NEW PS FROM DATA OF POSITION D AND CORRECTS PT                INT01820
      ALFDN=FALF-XPD                                                       INT01830
      CALL INTPLT(XA,YA,ZA,ALFDN,FPHI,NX,NY,CPD)                           INT01840
      ALFCN=FALF-XPC                                                       INT01850
      CALL INTPLT(XA,YA,ZA,ALFCN,FPHI,NX,NY,CPC)                           INT01860
905   WRITE(6,910) CPC,CPD                                                 INT01870
910   FORMAT(' NEW CPC & CPD CALLED',2F15.5)                               INT01880
      PSN=(PPC*CPD-PPD*CPC)/(CPD-CPC)                                      INT01890
907   WRITE(6,911) FALF,FPHI,PSN,PT                                        INT01900
911   FORMAT(' ITERATION VALUES',2F6.2,2F10.2)                             INT01910
C CONVERGENCE TESTS                                                        INT01920
950   DELPS=PSN-PS                                                         INT01930
951   WRITE(6,952) PS,PSN,DELPS                                            INT01940
952   FORMAT(' EPSPS VALUES',3F15.5)                                       INT01950
      EPSPS=ABS((PSN-PS)/PS)                                               INT01960
906   WRITE(6,961) EPSPS                                                   INT01970
961   FORMAT(' EPSPS IS' ,F25.15)                                          INT01980
      EPSPSC=.00000000099*(PT**2.0)-.00000098*PT                           INT01990
      IF(IEPS1.EQ.2) EPSPSC=.000000000099*(PT**2.0)-.0000098*PT            INT02000
      IF(IEPS2.EQ.2) EPSPSC=.000000000099*(PT**2.0)-.0000098*PT            INT02010
      IF(EPSPS.GT.EPSPSC) GO TO 310                                        INT02020
      GO TO 303                                                            INT02030
310   CONTINUE                                                             INT02040
      IIT=IIT+1                                                            INT02050
      IF(IIT.GT.10) GO TO 305                                              INT02060
      PS=PS+RELXPS*(PSN-PS)                                                INT02070
      PSC=PS                                                               INT02080
      PST=PS                                                               INT02090
924   WRITE(6,925) PS                                                      INT02100
```

```
FILE: WVOR     FORTRAN  P1                    N     P              S

525   FORMAT(' CURR PS IS ',F15.5)                                          INT02110
      GO TO 300                                                             INT02120
303   WRITE(6,171) FALF,FPPI,PSM,PT                                         INT02130
171   FORMAT(3F15.2,F15.3)                                                  INT02140
      GO TO 193                                                             INT02150
305   WRITE(6,307)                                                          INT02160
307   FORMAT(' SCANNED 100 TIMES')                                          INT02170
      GO TO 193                                                             INT02180
      END                                                                   INT02190
      SUBROUTINE INTSCS(ARP1,ARP2,ARLO,ARUP,ARIN,                           INT02200
     1HT1,HT2,RES1,RES2,X1,Y1,Z1,                                           INT02210
     2X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4,N1,N2)                                     INT02220
      DIMENSION RES1(10),RES2(10),ARRES(10),X1(40),Y1(40),                  INT02230
     1Z1(40,40),X2(40),Y2(40),Z2(40,40),X3(40),Y3(40),                      INT02240
     2Z3(40,40),X4(40),Y4(40),Z4(40,40)                                     INT02250
      RES1(1)=9000.                                                         INT02260
      RES1(2)=9000.                                                         INT02270
      RES2(1)=9000.                                                         INT02280
      RES2(2)=9000.                                                         INT02290
      ISL=1                                                                 INT02300
      ISR=1                                                                 INT02310
      DAR=1.                                                                INT02320
      AR2R=ARIN                                                             INT02330
      GO TO 152                                                             INT02340
150   AR2R=AR2R+DAR                                                         INT02350
      AR2RJ=AR2R                                                            INT02360
152   ARM1=AR2R-ARP1                                                        INT02370
      ARM2=AR2R-ARP2                                                        INT02380
C CHECK FOR LOWER CALIBRATION RANGE                                         INT02390
      IF(ARM1.LT.ARLO) GO TO 150                                            INT02400
      IF(ARM2.LT.ARLO) GO TO 150                                            INT02410
      K=1                                                                   INT02420
C CALCULATES INITIAL PENETRATION POINTS                                     INT02430
      CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)                            INT02440
      RAJL=ARRES(1)                                                         INT02450
      RAJR=ARRES(2)                                                         INT02460
      CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)                            INT02470
      RBJL=ARRES(1)                                                         INT02480
      RBJR=ARRES(2)                                                         INT02490
C CALCULATES SUCCESIVE PENETRATION POINTS                                   INT02500
      DO 140 I=1,10000                                                      INT02510
      AR2R=AR2R+DAR                                                         INT02520
      ARM1=AR2R-ARP1                                                        INT02530
      ARM2=AR2R-ARP2                                                        INT02540
C CHECK FOR UPPER CALIBRATION RANGE                                         INT02550
      IF(ARM1.GT.ARUP) GO TO 195                                            INT02560
      IF(ARM2.GT.ARUP) GO TO 195                                            INT02570
      CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)                            INT02580
      RAIL=ARRES(1)                                                         INT02590
      RAIR=ARRES(2)                                                         INT02600
      CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,AR42,ARRES)                            INT02610
      RBIL=ARRES(1)                                                         INT02620
      RBIR=ARRES(2)                                                         INT02630
      IF(RAJL.EQ.1000.) ISL=10                                              INT02640
      IF(REJL.EQ.1000.) ISL=10                                              INT02650
      IF(RBJR.EQ.1000.) ISR=10                                              INT02660
      IF(RAJR.EQ.1000.) ISR=10                                              INT02670
      IF(RAIL.EQ.1000.) ISL=10                                              INT02680
      IF(RBIL.EQ.1000.) ISL=10                                              INT02690
      IF(RBIR.EQ.1000.) ISR=10                                              INT02700
      IF(RAIR.EQ.1000.) ISR=10                                              INT02710
C CHECK FOR DIRECT HIT OR LEFT INTERSECTION,EVALUATE IF REQUIRED             INT02720
      IF(ISL.NE.10) GO TO 164                                               INT02730
      ISL=1                                                                 INT02740
      GO TO 183                                                             INT02750
164   IF(RAIL.NE.RBIL) GO TO 180                                            INT02760
      RES1(K)=RAIL                                                          INT02770
      RES2(K)=AR2R                                                          INT02780
      IF(K.EQ.2) GO TO 155                                                  INT02790
      K=2                                                                   INT02800
```

```
FILE: WVDP     FORTRAN  P1                    N     P           S
      GO TO 183                                                          INTC2810
C CHECK FOR INTERMEDIATE LEFT INTERSECTION,EVALUATE IF REQUIRED          INTC2820
180   IF(ISL.NE.10) GO TO 160                                            INTC2830
      ISL=1                                                              INTC2840
      GO TO 183                                                          INTC2850
160   IF((RAJL-RBJL)/(RAIL-RBIL).GT.0.0) GO TO 183                       INTC2860
      AA=(AR2RJ-AR2R)/(RAJL-RAIL)                                        INTC2870
      BA=AR2R-AA*RAIL                                                    INTC2880
      AB=(AR2RJ-AR2R)/(RBJL-RBIL)                                        INTC2890
      BB=AR2R-AB*RBIL                                                    INTC2900
      RES1(K)=(BB-BA)/(AA-AB)                                            INTC2910
      RES2(K)=AA*RES1(K)+BA                                              INTC2920
      IF(K.EQ.2) GO TO 155                                               INTC2930
      K=2                                                                INTC2940
C CHECK FOR DIRECT HIT OF RIGHT INTERSECTION,EVALUATE IF REQUIRED        INTC2950
183   IF(ISR.NE.10) GO TO 161                                            INTC2960
      ISR=1                                                              INTC2970
      GO TO 163                                                          INTC2980
161   IF(RAIR.NE.RBIR) GO TO 182                                         INTC2990
      RES1(K)=RBIR                                                       INTC3000
      RES2(K)=AR2R                                                       INTC3010
      IF(K.EQ.2) GO TO 155                                               INTC3020
      K=2                                                                INTC3030
      GO TO 163                                                          INTC3040
C CHECK FOR INTERMEDIATE RIGHT INTERSECTION,EVALUATE IF REQUIRED         INTC3050
182   IF(ISR.NE.10) GO TO 162                                            INTC3060
      ISR=1                                                              INTC3070
      GO TO 163                                                          INTC3080
162   IF((RAJR-RBJR)/(RAIR-RBIR).GT.0.0) GO TO 163                       INTC3090
      AA=(AR2RJ-AR2R)/(RAJR-RAIR)                                        INTC3100
      BA=AR2R-AA*RAIR                                                    INTC3110
      AB=(AR2RJ-AR2R)/(RBJR-RBIR)                                        INTC3120
      BB=AR2R-AB*RBIR                                                    INTC3130
      RES1(K)=(BB-BA)/(AA-AB)                                            INTC3140
      RES2(K)=AA*RES1(K)+BA                                              INTC3150
      IF(K.EQ.2) GO TO 155                                               INTC3160
      K=2                                                                INTC3170
C RE-INITIATE                                                            INTC3180
163   AR2RJ=AR2R                                                         INTC3190
      RAJL=RAIL                                                          INTC3200
      RAJR=RAIR                                                          INTC3210
      RBJL=RBIL                                                          INTC3220
      RBJR=RBIR                                                          INTC3230
140   CONTINUE                                                           INTC3240
      WRITE(6,210)                                                       INTC3250
210   FORMAT(' LOOP 140 THROUGH,NO POINTS FOUND')                        INTC3260
      GO TO 155                                                          INTC3270
195   IF(RES1(1).EQ.9000.0) GO TO 191                                    INTC3280
      IF(RES1(2).EQ.9000.0) GO TO 191                                    INTC3290
      IF(RES2(1).EQ.9000.0) GO TO 191                                    INTC3300
      IF(RES2(2).EQ.9000.0) GO TO 191                                    INTC3310
      GO TO 155                                                          INTC3320
C EVALUATION OF INTERSECTIONS WHEN CURVES ARE ALMOST TANGENT             INTC3330
191   YGC=0.0                                                            INTC3340
      CALL PENPTS(N2,N1,X3,Y3,Z3,HT1,YGC,ARRES)                          INTC3350
      R1=ABS(ARRES(2)-ARRES(1))/2                                        INTC3360
      CALL PENPTS(N2,N1,X4,Y4,Z4,HT2,YGC,ARRES)                          INTC3370
      R2=ABS(ARRES(2)-ARRES(1))/2                                        INTC3380
      RES2(1)=((R1**2)-(R2**2)+(ARD2**2))/(2.0*ARD2)                     INTC3390
      RES2(2)=RES2(1)                                                    INTC3400
      RES1(1)=SQRT(ABS((R1**2)-(RES2(1)**2)))                            INTC3410
      RES1(2)=-RES1(1)                                                   INTC3420
155   RETURN                                                             INTC3430
      END                                                                INTC3440
      SUBROUTINE PENPTS(NX,NY,X,Y,Z,HT,YG,XR)                            INTC3450
      DIMENSION Y(40),ZX(40),Z(40,40),X(40),XR(10)                       INTC3460
      IR=1                                                               INTC3470
C SEARCH FOR J OF LOWER Y LINE                                           INTC3480
      DO 101 I=1,NY                                                      INTC3490
      IF (Y(I).GE.YG) GO TO 102                                          INTC3500
```

```
FILE: WVDR      FORTRAN  P1                    N     P              S

101   CONTINUE                                                                INT03510
102   J=I-1                                                                   INT03520
C COARSE SCAN FOR ZERO PASS                                                   INT03530
C NEXT 4 LINES ARE EXECUTED IN FIRST SEARCH ONLY                              INT03540
      YC=(YC-Y(J))/(Y(J+1)-Y(J))                                              INT03550
      ZX(1)=(Z(1,J+1)-Z(1,J))*YC+Z(1,J)                                       INT03560
      IF(ZX(1).EQ.HT) GO TO 107                                               INT03570
      IB=2                                                                    INT03580
112   DO 103 I=IB,NX                                                          INT03590
      ZX(I)=(Z(I,J+1)-Z(I,J))*YC+Z(I,J)                                       INT03600
      ZM=(Z(I,J+1)-Z(I-1,J))*(YC-Y(J))/(Y(J+1)-Y(J))+Z(I-1,J)                 INT03610
      IF(ZX(I).EQ.HT) GO TO 108                                               INT03620
      IF((ZX(I-1)-HT)/(ZX(I)-HT).LT.0.0) GO TO 104                            INT03630
103   CONTINUE                                                                INT03640
      GO TO 116                                                               INT03650
C CALCULATION OF PENETRATIONS IN EITHER TRIANGLE                              INT03660
104   XM=X(I-1)+(X(I)-X(I-1))*(YC-Y(J))/(Y(J+1)-Y(J))                         INT03670
      IF(ZM.EQ.HT) GO TO 105                                                  INT03680
      IF((ZX(I)-HT)/(ZM-HT).LT.0.0) GO TO 106                                 INT03690
      DD=ABS(HT-ZM)+ABS(ZX(I-1)-HT)                                           INT03700
      XS=X(I-1)+(XM-X(I-1))*ABS(ZX(I-1)-HT)/DD                                INT03710
      GO TO 110                                                               INT03720
105   XS=XM+(X(I)-XM)*ABS(ZM-HT)/(ABS(ZM-HT)+ABS(ZX(I)-HT))                   INT03730
      GO TO 110                                                               INT03740
C LOGICAL PENETRATIONS ACCUMULATION                                           INT03750
107   XR(1)=X(I)                                                              INT03760
      I=1                                                                     INT03770
      GO TO 113                                                               INT03780
108   XR(IR)=X(I)                                                             INT03790
      GO TO 111                                                               INT03800
110   XR(IR)=XS                                                               INT03810
111   IF(IR.EQ.2) GO TO 114                                                   INT03820
113   IR=2                                                                    INT03830
      IB=I+1                                                                  INT03840
      GO TO 112                                                               INT03850
116   IF(IR.EQ.1) GO TO 117                                                   INT03860
      XR(2)=1000.0                                                            INT03870
      GO TO 114                                                               INT03880
117   XR(1)=1000.0                                                            INT03890
      XR(2)=1000.0                                                            INT03900
114   RETURN                                                                  INT03910
      END                                                                     INT03920
      SUBROUTINE INTPLT(X,Y,Z,XG,YG,NX,NY,ZRES)                               INT03930
C LINEAR INTERPOLATION OF CALIBRATION SURFACE TO EVALUATE                     INT03940
C CP VALUE AT CG-ORDINATES XG,YG. THE RESULT RETURNED IS ZRES                 INT03950
      DIMENSION X(40),Y(40),Z(40,40)                                          INT03960
C SEARCH FOR I OF LEFT X LINE                                                 INT03970
      DO 1 IC=1,NX                                                            INT03980
      IF(X(IC).GE.XG) GO TO 2                                                 INT03990
1     CONTINUE                                                                INT04000
2     I=IC-1                                                                  INT04010
C SEARCH FOR J OF LOWER Y LINE                                                INT04020
      DO 3 JC=1,NY                                                            INT04030
      IF(Y(JC).GE.YG) GO TO 4                                                 INT04040
3     CONTINUE                                                                INT04050
4     J=JC-1                                                                  INT04060
      AL=(Y(J+1)-Y(J))/(X(I+1)-X(I))                                          INT04070
      BL=Y(J)-AL*X(I)                                                         INT04080
      YCR=(AL*XG)+BL                                                          INT04090
      IF(YG.GE.YCR) GO TO 7                                                   INT04100
      IF(YG.LT.YCR) GO TO 5                                                   INT04110
C RESULTING Z IS IN LOWER TRIANGLE                                            INT04120
      X3=X(I+1)                                                               INT04130
      Y3=Y(J)                                                                 INT04140
      Z3=Z(I+1,J)                                                             INT04150
      GO TO 6                                                                 INT04160
C RASULTING Z IS IN UPPER TRIANGLE                                            INT04170
5     X3=X(I)                                                                 INT04180
      Y3=Y(J+1)                                                               INT04190
      Z3=Z(I,J+1)                                                             INT04200
```

```
FILE: WVDR     FORTRAN  P1                    N     P           S
C GENERAL CALCULATION:GOOD FOR BOTH TRIANGLES                                  INTC4210
6       X1=X(I)                                                                INTC4220
        Y1=Y(J)                                                                INTC4230
        Z1=Z(I,J)                                                              INTC4240
        X2=X(I+1)                                                              INTC4250
        Y2=Y(J+1)                                                              INTC4260
        Z2=Z(I+1,J+1)                                                          INTC4270
        AP=((Z1-Z3)*(Y1-Y2)-(Z1-Z2)*(Y1-Y3))/((X1-X3)*(Y1-Y2)-(X1-X2)          INTC4280
       1*(Y1-Y3))                                                              INTC4290
        BP=((Z1-Z2)-AP*(X1-X2))/(Y1-Y2)                                        INTC4300
        CP=Z1-AP*X1-BP*Y1                                                      INTC4310
        ZRES=AP*XG+BP*YG+CP                                                    INTC4320
        GO TO 100                                                              INTC4330
C RESULTING Z IS ON DIVIDING LINE                                              INTC4340
7       ZRES=(Z(I+1,J+1)-Z(I,J))*(XG-X(I))/(X(I+1)-X(I))+Z(I,J)                INTC4350
100     RETURN                                                                 INTC4360
        END                                                                    INTC4370
        SUBROUTINE MAXARY(Z,ZMAX,NX,NY)                                        INTC4380
C EVALUATION OF MAXIMUM OF OF CALIBRATION SURFACE MATRIX                       INTC4390
        DIMENSION Z(40,40)                                                     INTC4400
        DO 1 I=1,NX                                                            INTC4410
        DO 1 J=1,NY                                                            INTC4420
1       IF(Z(I,J).GE.ZMAX) ZMAX=Z(I,J)                                         INTC4430
        RETURN                                                                 INTC4440
        END                                                                    INTC4450
        SUBROUTINE MINARY(Z,ZMIN,NX,NY)                                        INTC4460
C EVALUATION OF MINIMUM OF OF CALIBRATION SURFACE MATRIX                       INTC4470
        DIMENSION Z(40,40)                                                     INTC4480
        DO 1 I=1,NX                                                            INTC4490
        DO 1 J=1,NY                                                            INTC4500
1       IF(Z(I,J).LE.ZMIN) ZMIN=Z(I,J)                                         INTC4510
        RETURN                                                                 INTC4520
        END                                                                    INTC4530
C END OF WVDR                                                                  INTC4540
                                                                               INTC4550
```

DISTRIBUTION LIST

		No. of Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Office of Research Administration Code 012A Naval Postgraduate School Monterey, California 93940	1
4.	Chairman Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	Director, Turbo-Propulsion Laboratory Department of Aeronautics Naval Postgraduate School Monterey, California 93940	30
6.	Dr. H. J. Mueller Research Administrator Code 310A Naval Air Systems Command Navy Department Washington, D. C. 20360	1
7.	Mr. Karl H. Guttmann Code 330C Naval Air Systems Command Navy Department Washington, D. C. 20360	1
8.	Mr. James R. Patton, Jr. Power Program, Code 473 Office of Naval Research Arlington, Virginia 22218	1
9.	Commanding Officer Naval Air Propulsion Test Center Attn: Mr. Vernon Lubosky Trenton, New Jersey 08628	1

10. National Aeronautics and Sapce Administration — 1
Lewis Research Center (Library)
2100 Brookpark Road
Cleveland, Ohio 44135

11. CAG Library — 1
The Boeing Company
Seattle, Washington 98124

12. Library — 1
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215

13. Library — 1
Pratt and Whitney Aircraft
Post Office Box 2691
West Palm Beach, Florida 33402

14. Library — 1
Pratt and Whitney Aircraft
East Hartford, Connecticut 06108

15. Chief, Fan and Compressor Branch — 1
Mail Stop 5-9
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

16. Director, Whittle Laboratory — 1
Department of Engineering
Cambridge University
ENGLAND

17. Prof. D. Adler — 10
Technion Israel Institute of Technology
Department of Mechanical Engineering
Haifa 32000
Israel

18. Prof. A. E. Breugelmans — 1
Institut von Karman de la Dynamique des Fluides
72 Chausee de Waterloo
1640 Rhode-St. Genese
Belgium

19. Mr. Robert O. Bullock — 1
Air Research Mfg. Corporation
Division of Garrett Corporation
402 South 36th Street
Phoenix, Arizona 85034

20. Dr. F. O. Carta — 1
United Technologies Research Labs
400 Main Street
Hartford, Connecticut 06108

21. Prof. Jacques Chauvin — 1
Universite D'Aix-Marseille
1 Rue Honnorat
Marseille, France

22. Mr. James V. Davis — 1
Teledyne CAE
1330 Laskey Road
Toledo, Ohio 43601

23. Mr. Jean Fabri — 1
ONERA
29, Ave. de la Division Leclerc
92 Chatillon
France

24. Prof. Dr. Ing Heinz E. Gallus — 1
Lehrstuhl und Institut fur Strahlantiebe und
Turbourbeitsmashinen
Rhein.-Westf. Techn. Hochschule Aachen
Templergraben 55
5100 Aachen, Germany

25. Professor J. P. Gostelow — 1
School of Mechanical Engineering
The New South Wales institute of Technology
Australia

26. DR. Ing. Hans-J. Heineman — 1
DFVLR-AVA
Bunsenstasse 10
3400 Gottingen, W. Germany

27. Prof. Ch. Hirsch — 1
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussells, Belgium

28. Prof. J. P. Johnston 1
Stanford University
Department of Mechanical Engineering
Stanford, California 94305

29. Prof. Jack L. Kerrebrock, Chairman 1
Aeronautics and Astronautics Department
31-265 Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

30. Dr. B. Lakshminarayana 1
Professor of Aerospace Engineering
The Pennsylvania State University
233 Hammond Building
University Park, Pennsylvania 16802

31. Mr. R. A. Langworthy 1
Army Aviation Material Laboratories
Department of the Army
Fort Eustis, Virginia 23604

32. Dr. A. A. Mikolajczak 1
Pratt and Whitney Aircraft
Engineering 2H
East Hartford, Connecticut 06108

33. Prof. Dr. L. G. Napolitano 1
Director
Institute of Aerodynamics
University of Naples
Viale C. Augusto
80125 Napoli
Italy

34. Prof. Erik Nilsson 1
Institutionen for Stromningsmaskinteknik
Chalmers Tekniska Hogskola
Fack, 402 20 Goteborg 5
Sweden

35. Prof. Gordon C. Oates 1
Department of Aeronautics and Astronautics
University of Washington
Seattle, Washington 98105

36. Prof. Walter F. O'Brian 1
Mechanical Engineering Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

37. Prof. Dr. K. Oswatitsch 1
Technische Hochschule
Karlsplatz 13
Vienna, Austria

38. Dr. P. A. Paranjee 1
Head, Propulsion Division
National Aeronautical Laboratory
Post Bag 1799
Bangalore - 17
India

39. R. E. Peacock 1
School of Mechanical Engineering
Cranfield Institute of Technology
Cranfield, Bedford MK43 0AL
ENGLAND

40. Dr. Bruce A. Reese 1
Director, Jet Propulsion Center
School of Mechanical Engineering
Purdue University
Layfayette, Indiana 47907

41. Dr. W. Schlachter 1
Brown, Boveri-Sulzer Turbomachinery Ltd
Dept. TDE
Escher Wyss Platz
Ch-8023 Zurich
Switzerland

42. Dr. George K. Serovy 1
Professor of Mechanical Engineering
208 Mechanical Engineering Building
Iowa State University
Ames, Iowa 50010

43. Dr. Fernando Sisto 1
Professor and Head of Mechanical Engineering Department
Stevens Institute of Technology
Castle Point, Hoboken, Jew Jersey 07030

44. Dr. Leroy H. Smith, Jr. 1
Manager, Compressor and Fan Technology Operation
General Electric Company
Aircraft Engine Technology Division
OTO Mail Drop H43
Cincinnati, Ohio 45215

45. Dr. W. Tabakoff 1
Professor, Department of Aerospace Engineering
University of Cincinnati
Cincinnati, Ohio 45221

46. Mr. P. Tramm 1
Manager, Research Labs
Detroit Diesel Allison Divison
General Motors
P. O. Box 894
Indianapolis, Indiana 46206

47. Prof. Dr. W. Traupel 1
Institut fur Thermische Turbomaschinen
Eidg. Technische Hochschule

48. Dr. Arthur J. Wennerstrom 1
ARL/LF
Wright-Patterson AFB
Dayton, Ohio 45433

49. Dr. H. Weyer 1
DFVLR
Linder Hohe
505 Porz-Wahn
Germany

50. Mr. P. F. Yaggy 1
Director
U. S. Army Aeronautical Research Laboratory
AMES Research Center
Moffett Field, California 94035